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

GAP Production Data Documentation

Bering Sea Survey Team[1,*] and Gulf of Alaska and Aleutian Island Survey Team[1,*]

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

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

Table of contents

List of Figures

List of Tables

Part I.

Welcome

Report run date: Wednesday, April 17, 2024

AFSC Bottom Trawl Surveys

AFSC bottom trawl surveys are conducted by the AFSC's Groundfish Assessment Program and Shellfish Assessment Program and are conducted in the Gulf of Alaska, Aleutian Islands, Eastern Bering Sea Slope, Eastern Bering Sea Shelf, and Northern Bering Sea. Each survey is a multispecies survey that collects data on the distribution, abundance, and biological characteristics of fish, crab, and other resources to inform groundfish stock assessment and management. These fishery-independent surveys are conducted in the summer aboard contracted commercial fishing vessels. Specifics regarding each of the surveys can be found below.

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.

Documentation Objective

As part of our commitment to open science, reproducibility, and transparency, we provide this metadata guide to compliment our public-domain data.

> 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 nmfs.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.

User Resources

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

Cite this data

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

- Design-Based Production Data (internal) (NOAA Fisheries Alaska Fisheries Science Center, Goundfish Assessment Program, 2024).

- AFSC RACE Groundfish Data for AKFIN (Alaska Fisheries Information Network (AKFIN), 2024).
- Public Data hosted on the Fisheries One Stop Shop (FOSS) Data Platform (NOAA Fisheries Alaska Fisheries Science Center, 2024).

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

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

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

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 and code of conduct 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.

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.

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 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.

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.

1. Survey Background

1.1. What we do

1.2. Who is conducting the research?

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

1.3. What is the research objective?

Learn more about the program. The objectives of these surveys are to:

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

1.4. 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.

1.5. Bottom trawl surveys and regions

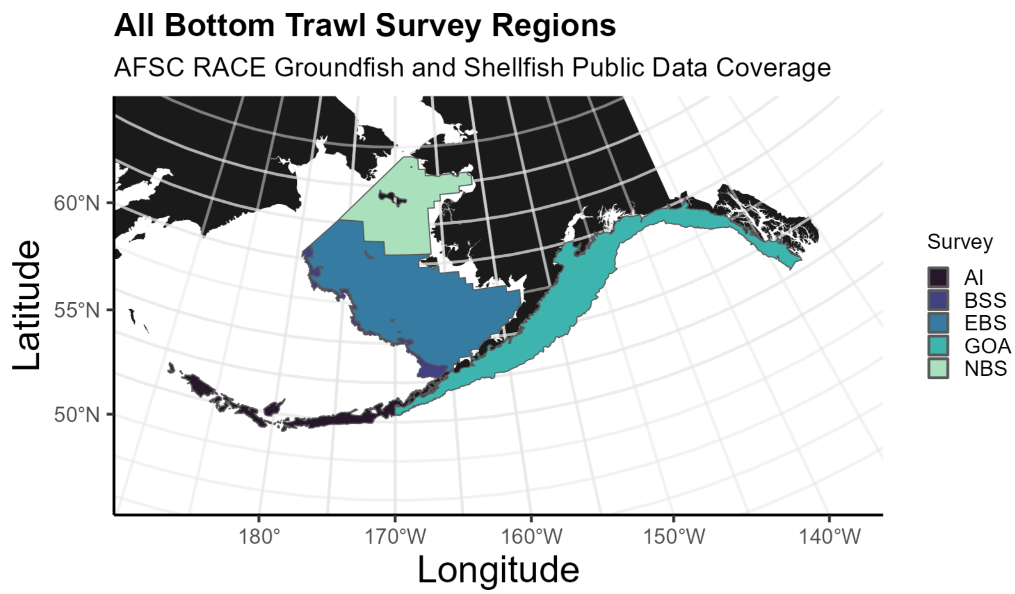


Figure 1.1.: Strata used in the all surveys.

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 (km2)	# Statistical Areas	# Possible Stations
Aleutian Islands Bottom Trawl Survey	52	2022 - 1991 (13)	1 - 500	64,415.0	80	1,312

Survey	Survey Definition ID	Years	Depth (m)	Area (km2)	# Statistical Areas	# Possible Stations
Eastern Bering Sea Slope Bottom Trawl Survey	78	2016 - 2002 (6)	201 - 1,200	32,861.3	37	
Eastern Bering Sea Crab/Groundfish Bottom Trawl Survey	98	2023 - 1982 (41)	1 - 200	492,989.	28	515
Gulf of Alaska Bottom Trawl Survey	47	2023 - 1990 (16)	1 - 1,000	313,784.9	37	6,939
Northern Bering Sea Crab/Groundfish Survey - Eastern Bering Sea Shelf Survey Extension	143	2023 - 2010 (6)	1 - 100	198,866.	4	144

1.5.1. Aleutian Islands

Most recent data report: (Von Szalay et al., 2023)

- 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.

1.5.2. Gulf of Alaska

Most recent data report: (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.

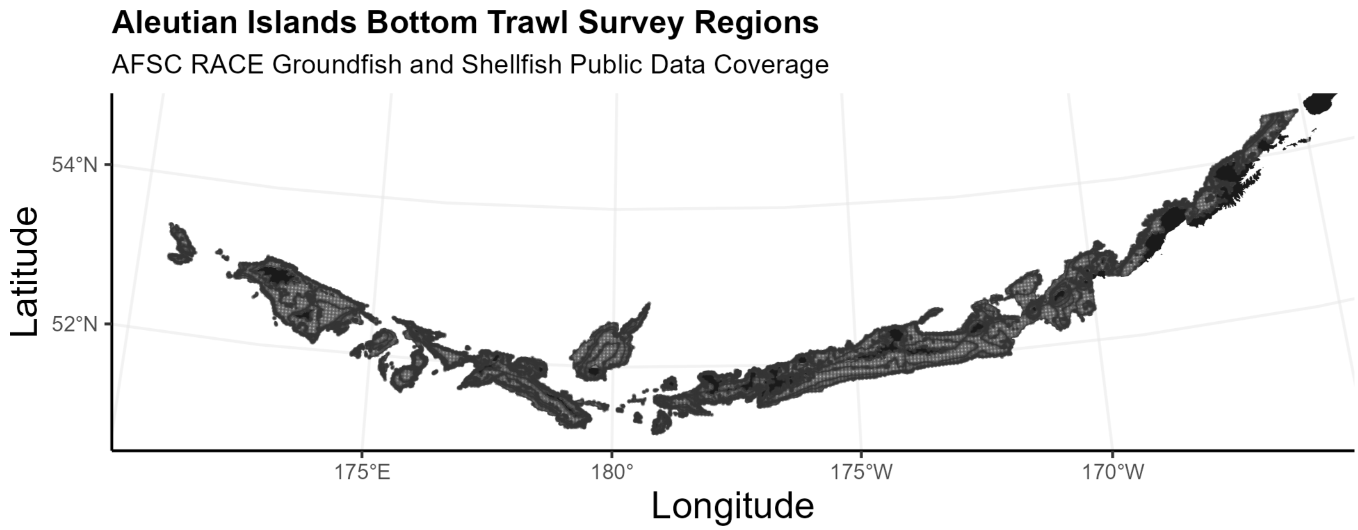


Figure 1.2.: Strata used in the Aleutian Islands bottom trawl survey.

1.5.3. Eastern Bering Sea Shelf

Most recent data report: (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.

1.5.4. Northern Bering Sea

Most recent data report: (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

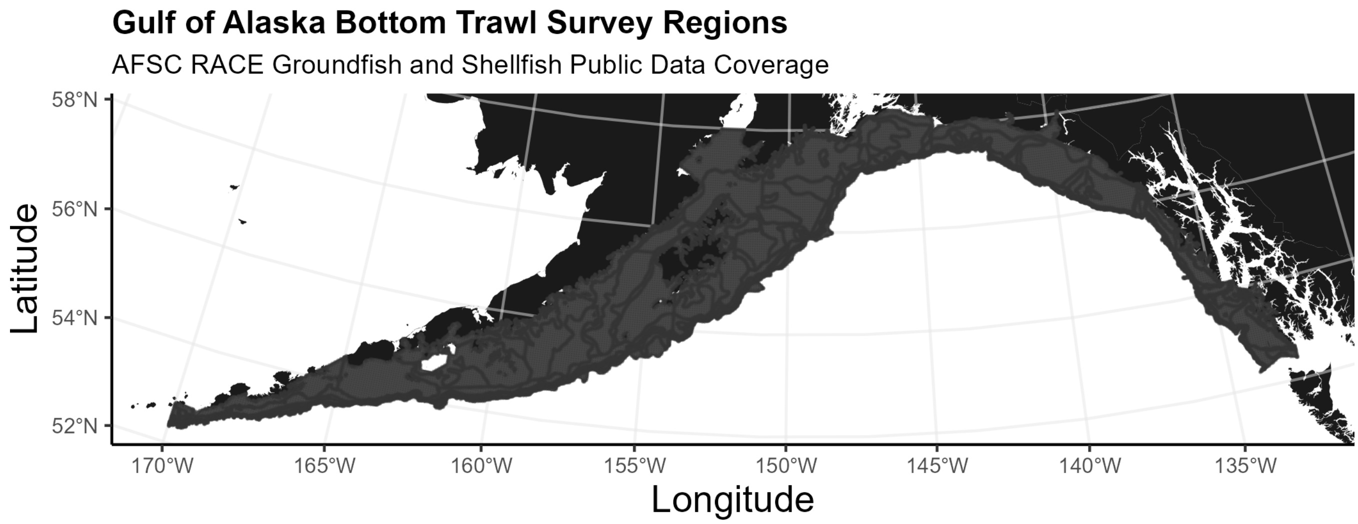


Figure 1.3.: Strata used in the Gulf of Alaska bottom trawl survey.

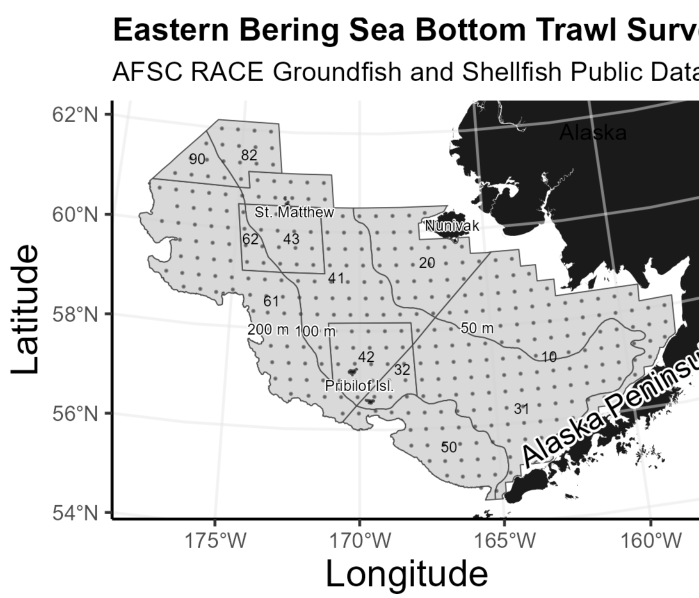


Figure 1.4.: Strata used in the Eastern Bering Sea bottom trawl survey.

- 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.

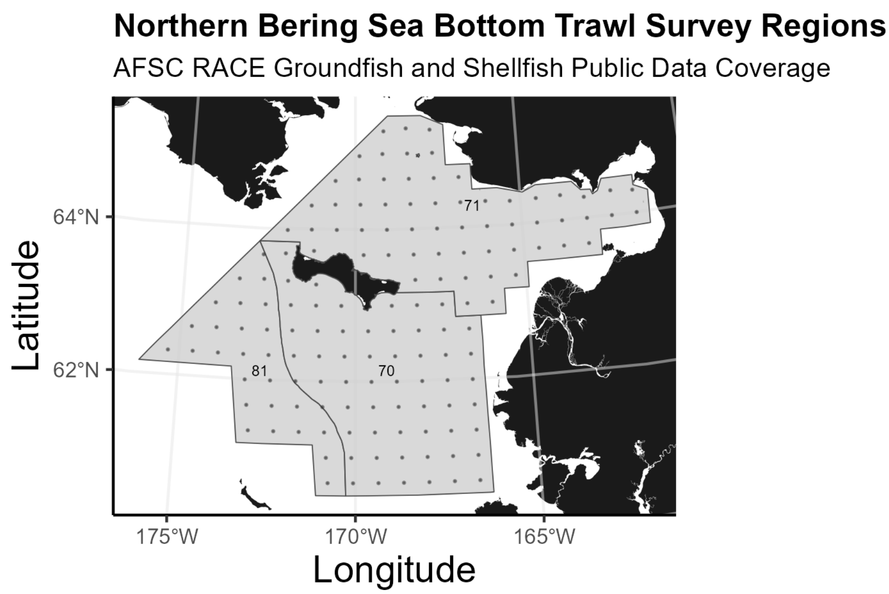


Figure 1.5.: Strata used in the Northern Bering Sea bottom trawl survey.

1.5.5. Eastern Bering Sea Upper Continental Slope

Most recent data report: (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.

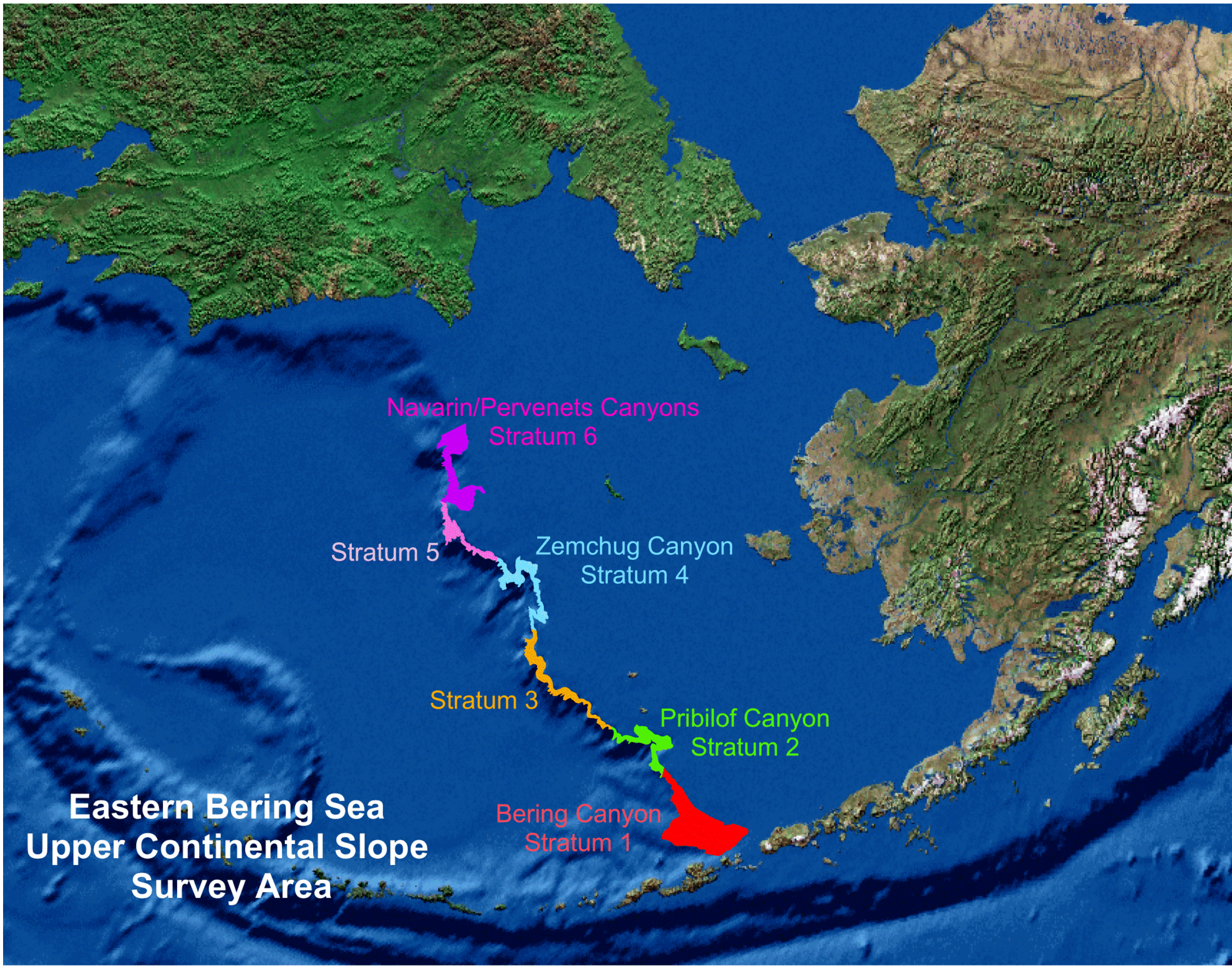


Figure 1.6.: Strata used in the Bering Sea Slope bottom trawl survey.

2. Workflow

2.1. Operational Product Development Timeline

Over the course of the year, the survey team is developing a variety of different data products. Planning and preparation for surveys happens in the late winter and spring, surveys occur in the summer, data validation takes place over the course of the survey and after the survey, and data products are produced through fall and late winter.

Table 2.1.: Operational product development timeline.

	January	February	March	April	May	June	July	August
Surveys								
Planning								
Developme								
Deployment (survey deliver-ables)								
Deployment (survey opera-tions)								
Triage (fixing bugs and errors)								

	January	February	March	April	May	June	July	August
User feedback and brain-storming								

2.2. Data workflow from boat to production

Organisms first need to be collected aboard the vessel before data can be entered into tablets.

The objective of this process is to take raw data, QA/QC and clean these data, curate standard data products for these survey. Please note, through this process we are not providing "data" (what we consider lower level data material; see the data levels section below) but "data products", which is intended to facilitate the most fool-proof standard interpretation of the data. These data products only use data from standard and validated hauls, and has undergone careful review.

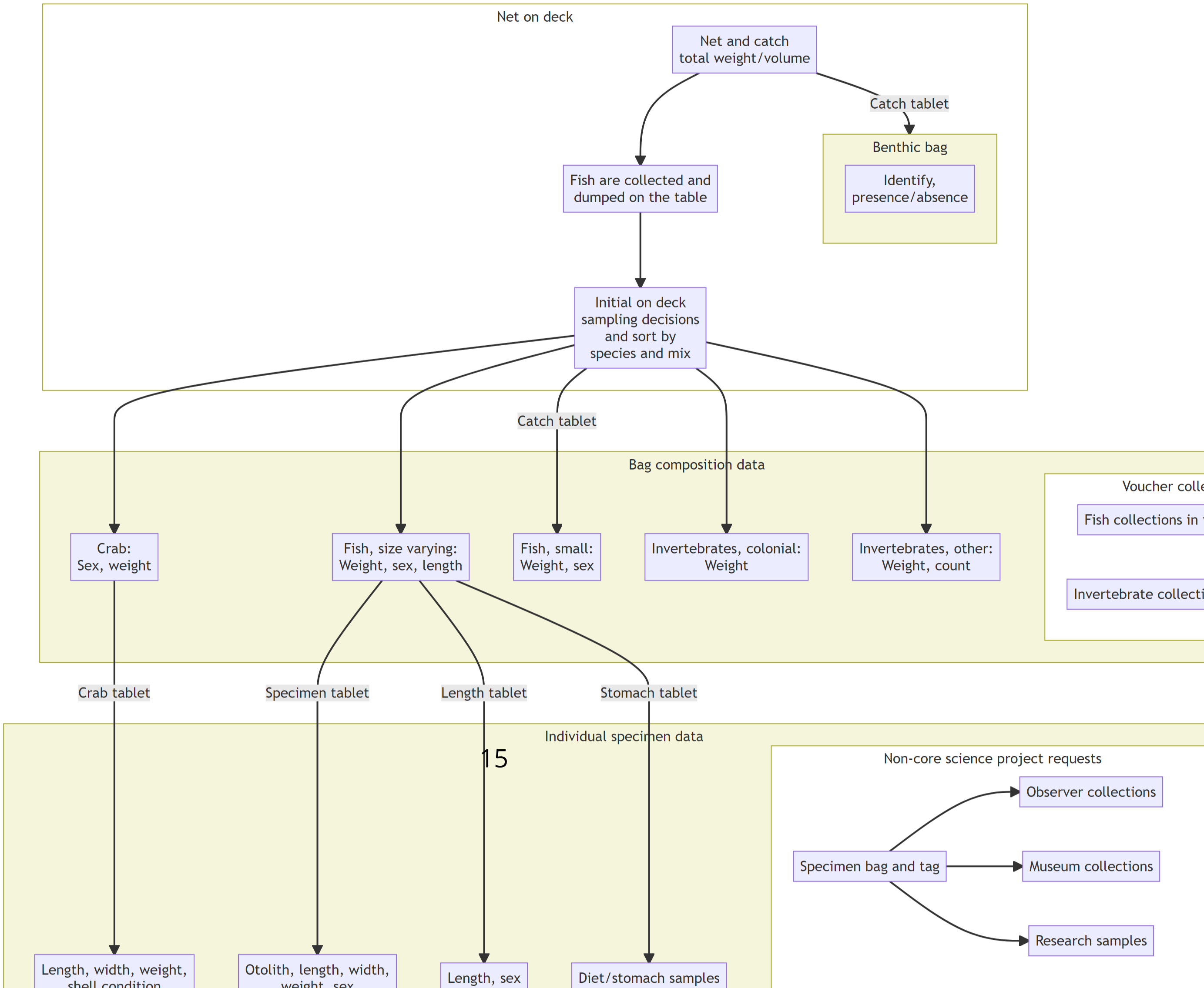
Once survey data collected on the vessel has been checked and validated, the `gap_products/code/run.R` script is used to orchestrate a sequence of programs that calculate the standard data products resulting from the NOAA AFSC GAP bottom trawl surveys. Standard data products are the CPUE, BIOMASS, SIZECOMP, 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.

> The data from these surveys constitute a **living data set** so we can continue to **provide the best available data to all partners, stakeholders, and fellow scientists**.

During each data product run cycle:

1. Versions of the tables in GAP_PRODUCTS are locally imported within the gap_products repository to compare with the updated production tables. Any

2. Workflow



Net on deck
Net and catch total weight/volume
Catch tablet
Benthic bag
Identify, presence/absence
Fish are collected and dumped on the table
Initial on deck sampling decisions and sort by species and mix
Catch tablet
Bag composition data
Crab: Sex, weight
Fish, size varying: Weight, sex, length
Fish, small: Weight, sex
Invertebrates, colonial: Weight
Invertebrates, other: Weight, count
Crab tablet
Specimen tablet
Length tablet
Stomach tablet
Individual specimen data
Non-core science project requests
Observer collections
Specimen bag and tag
Museum collections
Research samples
Diet/stomach samples
Length, sex

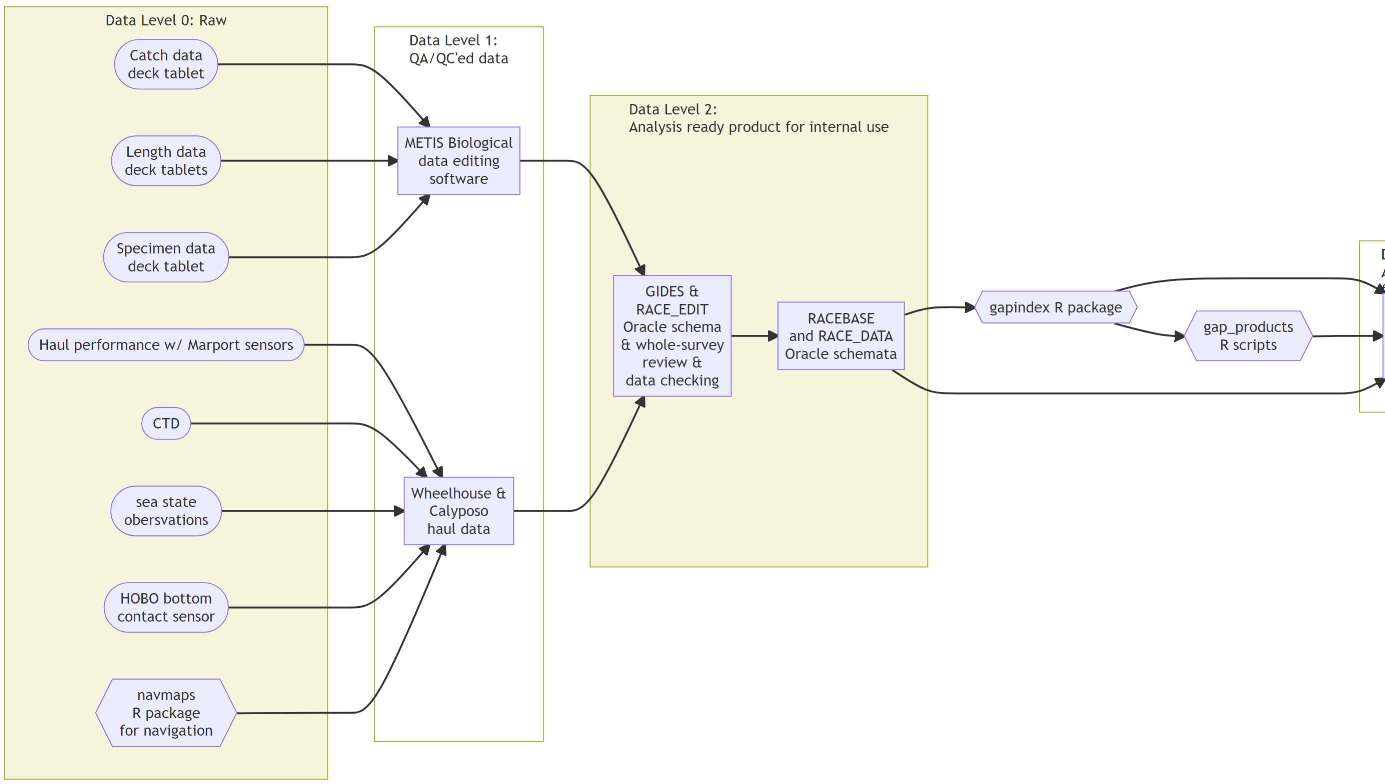


Figure 2.2.: Simplified data workflow from boat to production.

changes to a production table will be compared and checked to make sure those changes are intentional and documented.

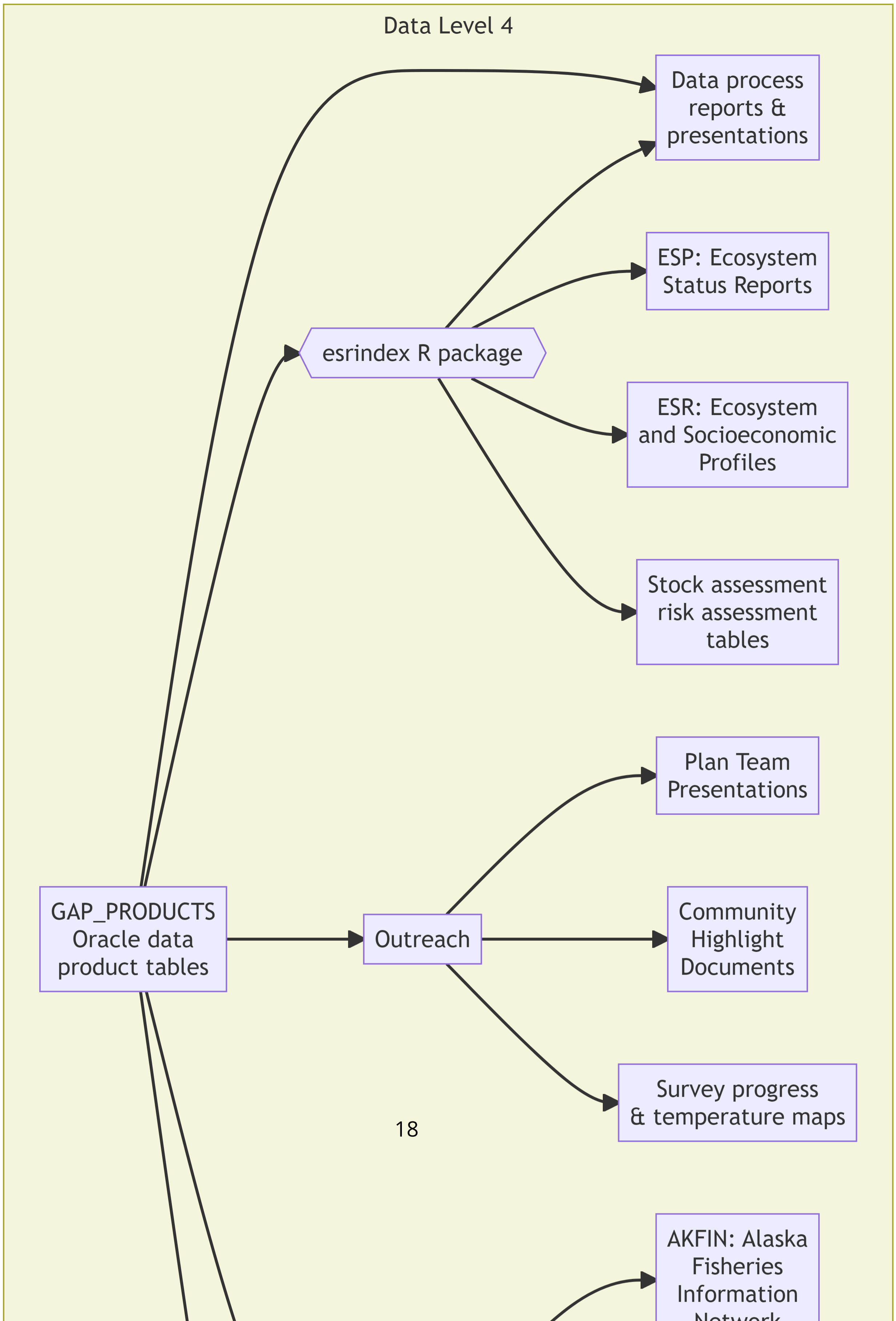
2. Use the `gapindex` R package to calculate the four major standard data products: CPUE, BIOMASS, SIZECOMP, AGECOMP. These tables are compared and checked to their respective locally saved copies and any changes to the tables are vetted and documented. These tables are then uploaded to the GAP_PRODUCTS Oracle schema.
3. Calculate the various materialized views for AKFIN and FOSS purposes. Since these are derivative of the tables in GAP_PRODUCTS as well as other base tables in RACEBASE and RACE_DATA, it is not necessary to check these views in addition to the data checks done in the previous steps.

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 1**: 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


Data Level 4
Data process reports & presentations
ESP: Ecosystem Status Reports
esrindex R package
ESR: Ecosystem and Socioeconomic Profiles
Stock assessment risk assessment tables
Plan Team Presentations
GAP_PRODUCTS Oracle data product tables
Outreach
Community Highlight Documents
Survey progress & temperature maps
AKFIN: Alaska Fisheries Information

- **Level 2**: Analysis-ready data products that are derived for a standardized extent and account for zeros and missing/bad data. Ex: CPUE tables, some data products in public-facing archives and repositories
- **Level 3**: Data products that are synthesized across a standardized extent, often inputs in a higher-level analytical product. Ex: Abundance indices, some data products in public-facing archives and repositories
- **Level 4**: Analytically generated data products that are derived from lower-level data, often to inform management. Ex: Biological reference points from stock assessments, Essential Fish Habitat layers, indicators in Ecosystem Status Reports and Ecosystem and Socioeconomic Profiles

3. News

3.1. News/change logs

– GAP_PRODUCTS ChangeLog (last produced on 2024-02-29) using gapindex v2.2.0: A new version of gapindex 2.2.0 was used for this production run and now accesses taxonomic information from RACEBASE.SPECIES instead of GAP_PRODUCTS.TAXONOMIC_CLASSIFICATION. As a result, there will be some SPECIES_CODE values that are supported due to slight differences between the two tables. Discussion in this github issue #54. As a result there are new cpue records for SPECIES_CODE values 22290 and 22292 and removed cpue records for SPECIES_CODE values 21345, 22200 and 69326.

– GAP_PRODUCTS ChangeLog (last produced on 2024-01-09) using gapindex v2.1.3: A new version of gapindex (v2.1.3) was used to produced these data. Data for SPECIES_CODE 68590 (Chionoecetes hybrids) are now removed, per this issue (https://github.com/afsc-gap-products/gap_products/issues/3). New read otolith data were incorporated into the age compositions. GOA depth subareas are now included in the size comps, and there were some modifications with EBS skate length data that are now incorporated into the length compositions.

– GAP_PRODUCTS ChangeLog (last produced on 2023-11-17) using gapindex v2.1.2: A new version of gapindex (v2.1.2) was used to produced these data. There was a slight change to how subarea biomass totals are calculated that was not fully addressed in v2.1.1. The modified biomass records reflect this change.

– GAP_PRODUCTS ChangeLog (last produced on 2023-11-14) using gapindex v2.1.1: A new version of gapindex (v2.1.1) was used to produced these data. There was a slight change to how subarea biomass totals are calculated. The modified biomass records reflect this change. New 2022 otolith data were available since the last iteration of the GAP_PRODUCTS for Aleutian Island Pacific ocean perch and northern rockifsh and Eastern Bering Sea northern rock sole. Zero-filled CPUE records for four GOA species codes (SPECIES_CODE: 21210, 30010, 30360, 77102, 98101) were added due to how the 1990 data were integrated in the last production run of GAP_PRODUCTS. Two Arctic cod (SPECIES_CODE: 21725) and one plain sculpin (SPECIES_CODE:

21371) count records were modified in the NBS data, which changes the numerical CPUE estimates for those hauls which changes the estimated population abundance and size composition for those species.

– Groundfish Assessment Program Survey Data Serving and Data Improvements: Initial data changes brief distributed to SSMA and other partners by Ned Laman, Zack Oyafuso, and Emily Markowitz

– Run 2023-06-01 gapindex v2.1.0: Initial compiling and planning notes

4. Code of Conduct

4.1. What are Codes of Conduct?

Codes of Conduct are voluntary sets of rules that assist creators, developers, and users of code and data with data protection compliance and accountability in specific sectors or relating to particular processing operations.

Codes can help organizations to ensure all participants follow best practices and rules designed specifically for their sector or processing operations, thus enhancing compliance and collaboration. They are developed and managed by an association or other body (the 'Code Owner') which is representative of a sector (or category of data controllers or processors), with the expert and sectoral knowledge of how to enhance data protection in their area.

4.1.1. Code of Conduct from the nmfs-opensci GitHub.

5. NOAA Fisheries Open Science Code of Conduct

This code of conduct was developed and adapted from the Atom code of conduct in October 2021.

5.1. Our Pledge

In the interest of fostering an open and welcoming environment, we as contributors and maintainers pledge to making participation in our project and our community a harassment-free experience for everyone, regardless of age, body size, disability, ethnicity, gender identity and expression, level of experience, nationality, personal appearance, race, religion, or sexual identity and orientation.

5.2. Our Standards

Examples of behavior that contributes to creating a positive environment include:

- Using welcoming and inclusive language
- Being respectful of differing viewpoints and experiences
- Gracefully accepting constructive criticism
- Focusing on what is best for the community
- Showing empathy towards other community members

Examples of unacceptable behavior by participants include:

- The use of sexualized language or imagery and unwelcome sexual attention or advances
- Trolling, insulting/derogatory comments, and personal or political attacks
- Public or private harassment

- Publishing others' private information, such as a physical or electronic address, without explicit permission
- Other conduct which could reasonably be considered inappropriate in a professional setting

5.3. Our Responsibilities

Project maintainers are responsible for clarifying the standards of acceptable behavior and are expected to take appropriate and fair corrective action in response to any instances of unacceptable behavior.

Project maintainers have the right and responsibility to remove, edit, or reject comments, commits, code, wiki edits, issues, and other contributions that are not aligned to this Code of Conduct, or to ban temporarily or permanently any contributor for other behaviors that they deem inappropriate, threatening, offensive, or harmful.

5.4. Scope

This Code of Conduct applies both within project spaces and in public spaces when an individual is representing the project or its community. Examples of representing a project or community include using an official project e-mail address, posting via an official social media account, or acting as an appointed representative at an online or offline event. Representation of a project may be further defined and clarified by project maintainers.

5.5. Enforcement

Instances of abusive, harassing, or otherwise unacceptable behavior may be reported by contacting the project team. All complaints will be reviewed and investigated and will result in a response that is deemed necessary and appropriate to the circumstances. Further details of specific enforcement policies may be posted separately.

5.6. Attribution

This Code of Conduct is adapted from the Contributor Covenant, version 1.4, available at https://contributor-covenant.org/version/1/4

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, 2024). Add "note = {Accessed: mm/dd/yyyy}" to append the day this data was accessed.

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

6. Data description

6.1. Data tables

6.1.1. AGECOMP

Region-level age compositions by sex/length bin.

Number of rows: 662,289

Number of columns: 10

Column name from data

Descriptive column Name

Units

Oracle data type

Column description

AGE

Taxon age bin (yrs)

integer

NUMBER(38,0)

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

AREA_ID

Area ID

ID key code

NUMBER(38,0)

Area ID key 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_ID_FOOTPRINT

NA

NA

NA

NA

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(38,0)

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

SEX

Sex of a specimen

ID key code

NUMBER(38,0)

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

SPECIES_CODE

Taxon code

ID key 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 key code

NUMBER(38,0)

The survey definition ID key code uniquely identifies a survey/survey design. Integer code that uniquely identifies survey. Full list of survey definition IDs are in RACE_-DATA.SURVEY_DEFINITIONS. IDs used in GAP_PRODUCTS are: 47 (Gulf of Alaska); 52 (Aleutian Islands); 78 (Bering Sea Slope); 98 (Eastern Bering Sea Shelf); 143 (Northern Bering Sea Shelf). The column ‘survey_definition_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)

Year the observation (survey) was collected.

6.1.2. AREA

This table contains all of the information related to the various strata, subareas, INPFC and NMFS management areas, and regions for the Aleutian Islands, Gulf of Alaska, and Bering Sea shelf and slope bottom trawl surveys. These tables are created by the Resource Assessment and Conservation Engineering Division (RACE) Groundfish Assessment Program (GAP) of the Alaska Fisheries Science Center (AFSC). The GitHub repository for the scripts that created this code can be found at https://github.com/afsc-gap-products/gap_products. These data were last updated

April 02, 2024. There are no legal restrictions on access to the data. For more information about codes used in the tables, please refer to the survey code books (https://www.fisheries.noaa.gov/resource/document/groundfish-survey-species-code-manual-and-data-codes-manual).

Number of rows: 395

Number of columns: 9

Column name from data

Descriptive column Name

Units

Oracle data type

Column description

AREA_ID

Area ID

ID key code

NUMBER(38,0)

Area ID key 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.

AREA_TYPE

Area ID type description

category

VARCHAR2(255 BYTE)

The type of stratum that AREA_ID represents. Types include: STRATUM (the smallest building-block unit of area in these surveys), REGION, DEPTH, SUBAREA, INPFC BY DEPTH, INPFC, SUBAREA BY DEPTH, REGULATORY AREA, NMFS STATISTICAL AREA.

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)

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 key code

NUMBER(38,0)

The survey definition ID key code uniquely identifies a survey/survey design. Integer code that uniquely identifies survey. Full list of survey definition IDs are in RACE_-DATA.SURVEY_DEFINITIONS. IDs used in GAP_PRODUCTS are: 47 (Gulf of Alaska); 52 (Aleutian Islands); 78 (Bering Sea Slope); 98 (Eastern Bering Sea Shelf); 143 (Northern Bering Sea Shelf). The column 'survey_definition_id' is associated with the 'srvy' and 'survey' columns. For a complete list of surveys, review the code books.

6.1.3. BIOMASS

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

Number of rows: 2,576,866

Number of columns: 16

Column name from data

Descriptive column Name

Units

Oracle data type

Column description

AREA_ID

Area ID

ID key code

NUMBER(38,0)

Area ID key 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

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

Valid hauls

count

NUMBER(38,0)

Total number of hauls.

N_LENGTH

Hauls with taxon lengths

count

NUMBER(38,0)

Total number of hauls with length data.

N_WEIGHT

Hauls with catch

count

NUMBER(38,0)

Total number of hauls with positive catch biomass.

POPULATION_COUNT

Estimated population

numeric

NUMBER(38,0)

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

POPULATION_VAR

Estimated population variance

numeric

NUMBER(38,6)

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

SPECIES_CODE

Taxon code

ID key 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 key code

NUMBER(38,0)

The survey definition ID key code uniquely identifies a survey/survey design. Integer code that uniquely identifies survey. Full list of survey definition IDs are in RACE_-DATA.SURVEY_DEFINITIONS. IDs used in GAP_PRODUCTS are: 47 (Gulf of Alaska); 52 (Aleutian Islands); 78 (Bering Sea Slope); 98 (Eastern Bering Sea Shelf); 143 (Northern Bering Sea Shelf). The column 'survey_definition_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)

Year the observation (survey) was collected.

6.1.4. CPUE

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

Number of rows: 21,360,338

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 or samples collected.

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 key code

NUMBER(38,0)

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

SPECIES_CODE

Taxon code

ID key 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

Sample or taxon weight (kg)

kilograms

NUMBER(38,3)

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

6.1.5. SURVEY_DESIGN

This table contains for a given survey (via SURVEY_DEFINITION_ID) and survey year (YEAR), which version (DESIGN_YEAR) of the AREA_IDs that were used to calculate the various standard data products. These tables are created by the Resource Assessment and Conservation Engineering Division (RACE) Groundfish Assessment Program (GAP) of the Alaska Fisheries Science Center (AFSC). The GitHub repository for the scripts that created this code can be found at https://github.com/afsc-gap-products/gap_products. These data were last updated April 02, 2024. There are no legal restrictions on access to the data. For more information about codes used in the tables, please refer to the survey code books (https://www.fisheries.noaa.gov/resource/document/groundfish-survey-species-code-manual-and-data-codes-manual).

Number of rows: 87

Number of columns: 3

Column name from data

Descriptive column Name

Units

Oracle data type

Column description

DESIGN_YEAR

Design year

year

NUMBER(10,0)

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 key code

NUMBER(38,0)

The survey definition ID key code uniquely identifies a survey/survey design. Integer code that uniquely identifies survey. Full list of survey definition IDs are in RACE_-DATA.SURVEY_DEFINITIONS. IDs used in GAP_PRODUCTS are: 47 (Gulf of Alaska); 52 (Aleutian Islands); 78 (Bering Sea Slope); 98 (Eastern Bering Sea Shelf); 143 (Northern Bering Sea Shelf). The column 'survey_definition_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)

Year the observation (survey) was collected.

6.1.6. METADATA_TABLE

These columns provide the table metadata for all of the tables and views in GAP_-PRODUCTS. These tables are created by the Resource Assessment and Conservation Engineering Division (RACE) Groundfish Assessment Program (GAP) of the Alaska Fisheries Science Center (AFSC). The GitHub repository for the scripts that created this code can be found at https://github.com/afsc-gap-products/gap_products. These data were last updated April 02, 2024. There are no legal restrictions on access to the data. For more information about codes used in the tables, please refer to the survey code books (https://www.fisheries.noaa.gov/resource/document/groundfish-survey-species-code-manual-and-data-codes-manual).

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

VARCHAR2(4000 BYTE)

Table metadata sentence.

METADATA_SENTENCE_NAME

Metadata sentence name

text

VARCHAR2(4000 BYTE)

Name of table metadata sentence.

METADATA_SENTENCE_TYPE

Sentence type

text

VARCHAR2(4000 BYTE)

Type of sentence to have in table metadata.

6.1.7. STRATUM_GROUPS

This table contains all of strata that are contained within a given subarea, INPFC or NMFS management area, or region for the Aleutian Islands, Gulf of Alaska, and Bering Sea shelf and slope bottom trawl surveys. These tables are created by the Resource Assessment and Conservation Engineering Division (RACE) Groundfish Assessment Program (GAP) of the Alaska Fisheries Science Center (AFSC). The GitHub repository for the scripts that created this code can be found at https://github.com/afsc-gap-products/gap_products. These data were last updated April 02, 2024. There are no legal restrictions on access to the data. For more information about codes used in the tables, please refer to the survey code books (https://www.fisheries.noaa.gov/resource/document/groundfish-survey-species-code-manual-and-data-codes-manual).

Number of rows: 768

Number of columns: 4

Column name from data

Descriptive column Name

Units

Oracle data type

Column description

AREA_ID

Area ID

ID key code

NUMBER(38,0)

Area ID key 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 key 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 key code

NUMBER(38,0)

The survey definition ID key code uniquely identifies a survey/survey design. Integer code that uniquely identifies survey. Full list of survey definition IDs are in RACE_-DATA.SURVEY_DEFINITIONS. IDs used in GAP_PRODUCTS are: 47 (Gulf of Alaska); 52

(Aleutian Islands); 78 (Bering Sea Slope); 98 (Eastern Bering Sea Shelf); 143 (Northern Bering Sea Shelf). The column 'survey_definition_id' is associated with the 'srvy' and 'survey' columns. For a complete list of surveys, review the code books.

6.1.8. SIZECOMP

Stratum/subarea/region-level size compositions by sex. This table was created by the Resource Assessment and Conservation Engineering Division (RACE) Groundfish Assessment Program (GAP) of the Alaska Fisheries Science Center (AFSC). There are legal restrictions on access to the data. These data are not intended for public dissemination and should not be shared without the explicit written consent of the data managers and owners (NOAA Fisheries). The GitHub repository for the scripts that created this code can be found at https://github.com/afsc-gap-products/gap_products. For more information about codes used in the tables, please refer to the survey code books (https://www.fisheries.noaa.gov/resource/document/groundfish-survey-species-code-manual-and-data-codes-manual). These data were last updated April 10, 2024.

Number of rows: 3,209,517

Number of columns: 7

Column name from data

Descriptive column Name

Units

Oracle data type

Column description

AREA_ID

Area ID

ID key code

NUMBER(38,0)

Area ID key 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(38,0)

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

SEX

Sex of a specimen

ID key code

NUMBER(38,0)

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

SPECIES_CODE

Taxon code

ID key 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 key code

NUMBER(38,0)

The survey definition ID key code uniquely identifies a survey/survey design. Integer code that uniquely identifies survey. Full list of survey definition IDs are in RACE_-DATA.SURVEY_DEFINITIONS. IDs used in GAP_PRODUCTS are: 47 (Gulf of Alaska); 52 (Aleutian Islands); 78 (Bering Sea Slope); 98 (Eastern Bering Sea Shelf); 143 (Northern Bering Sea Shelf). The column ‘survey_definition_id’ is associated with the ‘srvy’ and ‘survey’ columns. For a complete list of surveys, review the code books.

6. Data description

YEAR

Survey year

year

NUMBER(10,0)

Year the observation (survey) was collected.

Part III.

AKFIN

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

The Alaska Fisheries Information Network

The Alaska Fisheries Information Network (AKFIN) is a regional program that consolidates and supports the processing, analysis, and reporting of fisheries data for Alaskan fisheries. AKFIN integrates this information into a single data management system using consistent methods and standardized formats. The resulting data enables fishery managers, scientists, and associated agencies to supervise fisheries resources more effectively and efficiently. The AKFIN database contains much of the data needed to complete stock assessments, including GAP trawl survey data. .

Data Access Options

Direct database connection If you are an AFSC employee you may access the AKFIN oracle database directly while on the NOAA network or VPN. Note that this is a separate database from the AFSC oracle database referenced above, and requires separate credentials. If you do not already have an AKFIN account you can request one here. NOAA IT will need to add AKFIN access to your tnsnames.ora file (They do this frequently). Once your connection is established data may be accessed through SQL queries using SQL developer, R, or python.

AKFIN Answers

(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 AKFIN direct database credentials. The distribution of GAP_PRODUCTS on AKFIN Answers is planned but not yet implemented. The RACE Survey tab on the stock assessment dashboard contains reports generated from now depreciated tables that predated the GAP_PRODUCTS tables. AKFIN will keep these reports for reference but they will not be updated 2024 onward.

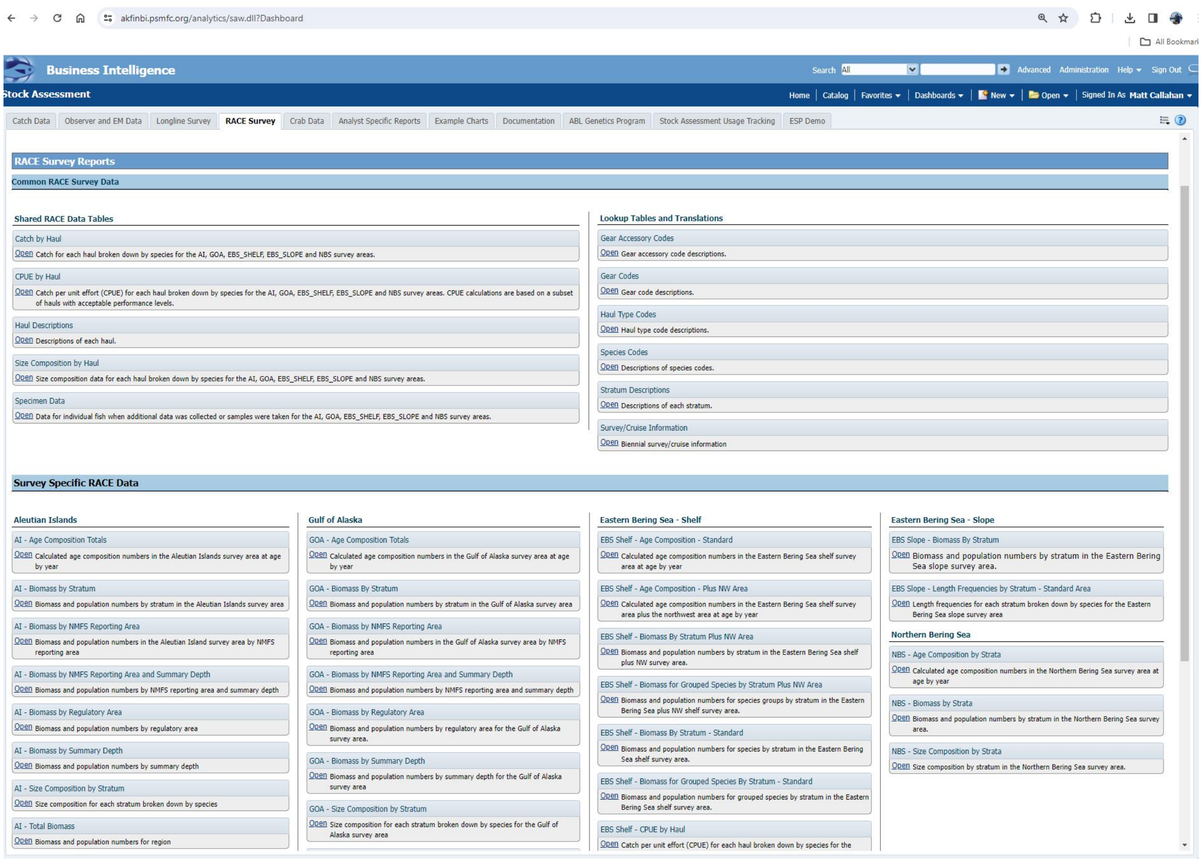


Figure 6.1.: AKFIN platfrom.

Web Service

AKFIN has developed web services (apis) to distribute GAP data. Like the GAP_PRODUCTS schema, these are under active development. These do not require VPN or an oracle connection but they are protected by Oracle authentication, please contact matt.callahan@noaa.gov for information on how to get an api token to use this option.

The url structure is "https://apex.psmfc.org/akfin/data_marts/gap_products/gap_[base table name]" . For example "https://apex.psmfc.org/akfin/data_marts/gap_products/gap_biomass" is the base url to get data from the akfin_biomass table. Web services linked to large tables have mandatory parameters to reduce data download size. For example to get agecomp data for Bering Sea pollock in area_id 10 in 2022 you would use "https://apex.psmfc.org/akfin/data_marts/gap_products/gap_biomass?survey_definition_id=98&area_id=10&species_code=21740&start_year=2022&end_year=2022".

If you're using R to pull data through web services you might find the akfingapdata (pronounced akfin-gap-data not ak-eff-ing-app-data) R package helpful.

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), 2024). Add "note = {Accessed: mm/dd/yyyy}" to append the day this data was accessed.

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

7. Data description

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.

7.1. Data tables

7.1.1. AKFIN_AGECOMP

This table is a copy of GAP_PRODUCTS.AGECOMP and does not have any other object dependencies. These data are produced by the Resource Assessment and Conservation Engineering Division (RACE) Groundfish Assessment Program (GAP) of the Alaska Fisheries Science Center (AFSC). There are legal restrictions on access to the data. These data are not intended for public dissemination and should not be shared without the explicit written consent of the data managers and owners (NOAA Fisheries). The GitHub repository for the scripts that created this code can be found at https://github.com/afsc-gap-products/gap_products. These data were last updated April 10, 2024.

Number of rows: 662,289

Number of columns: 10

Column name from data

Descriptive column Name

Units

Oracle data type

Column description

AGE

Taxon age bin (yrs)

integer

NUMBER(38,0)

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

AREA_ID

Area ID

ID key code

NUMBER(38,0)

Area ID key 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_ID_FOOTPRINT

NA

NA

NA

NA

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(38,0)

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

SEX

Sex of a specimen

ID key code

NUMBER(38,0)

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

SPECIES_CODE

Taxon code

ID key 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 key code

NUMBER(38,0)

The survey definition ID key code uniquely identifies a survey/survey design. Integer code that uniquely identifies survey. Full list of survey definition IDs are in RACE_-DATA.SURVEY_DEFINITIONS. IDs used in GAP_PRODUCTS are: 47 (Gulf of Alaska); 52 (Aleutian Islands); 78 (Bering Sea Slope); 98 (Eastern Bering Sea Shelf); 143 (Northern Bering Sea Shelf). The column ‘survey_definition_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)

Year the observation (survey) was collected.

7.1.2. AKFIN_AREA

This table is a copy of GAP_PRODUCTS.AREA and does not have any other object dependencies. These data are produced by the Resource Assessment and Conservation Engineering Division (RACE) Groundfish Assessment Program (GAP) of the Alaska Fisheries Science Center (AFSC). There are legal restrictions on access to the data. These data are not intended for public dissemination and should not be shared without the explicit written consent of the data managers and owners (NOAA Fisheries). The GitHub repository for the scripts that created this code can be found at https://github.com/afsc-gap-products/gap_products. These data were last updated April 10, 2024.

Number of rows: 395

Number of columns: 9

Column name from data

Descriptive column Name

Units

Oracle data type

Column description

AREA_ID

Area ID

ID key code

NUMBER(38,0)

Area ID key 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.

AREA_TYPE

Area ID type description

category

VARCHAR2(255 BYTE)

The type of stratum that AREA_ID represents. Types include: STRATUM (the smallest building-block unit of area in these surveys), REGION, DEPTH, SUBAREA, INPFC BY DEPTH, INPFC, SUBAREA BY DEPTH, REGULATORY AREA, NMFS STATISTICAL AREA.

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)

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 key code

NUMBER(38,0)

The survey definition ID key code uniquely identifies a survey/survey design. Integer code that uniquely identifies survey. Full list of survey definition IDs are in RACE_-DATA.SURVEY_DEFINITIONS. IDs used in GAP_PRODUCTS are: 47 (Gulf of Alaska); 52 (Aleutian Islands); 78 (Bering Sea Slope); 98 (Eastern Bering Sea Shelf); 143 (Northern Bering Sea Shelf). The column 'survey_definition_id' is associated with the 'srvy' and 'survey' columns. For a complete list of surveys, review the code books.

7.1.3. AKFIN_BIOMASS

This table is a copy of GAP_PRODUCTS.BIOMASS and does not have any other object dependencies. These data are produced by the Resource Assessment and Conservation Engineering Division (RACE) Groundfish Assessment Program (GAP) of the Alaska Fisheries Science Center (AFSC). There are legal restrictions on access to the data. These data are not intended for public dissemination and should not be shared without the explicit written consent of the data managers and owners (NOAA Fisheries). The GitHub repository for the scripts that created this code can be found at https://github.com/afsc-gap-products/gap_products. These data were last updated April 10, 2024.

Number of rows: 2,576,866

Number of columns: 16

Column name from data

Descriptive column Name

Units

Oracle data type

Column description

AREA_ID

Area ID

ID key code

NUMBER(38,0)

Area ID key 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

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

Valid hauls

count

NUMBER(38,0)

Total number of hauls.

N_LENGTH

Hauls with taxon lengths

count

NUMBER(38,0)

Total number of hauls with length data.

N_WEIGHT

Hauls with catch

count

NUMBER(38,0)

Total number of hauls with positive catch biomass.

POPULATION_COUNT

Estimated population

numeric

NUMBER(38,0)

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

POPULATION_VAR

Estimated population variance

numeric

NUMBER(38,6)

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

SPECIES_CODE

Taxon code

ID key 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 key code

NUMBER(38,0)

The survey definition ID key code uniquely identifies a survey/survey design. Integer code that uniquely identifies survey. Full list of survey definition IDs are in RACE_-DATA.SURVEY_DEFINITIONS. IDs used in GAP_PRODUCTS are: 47 (Gulf of Alaska); 52 (Aleutian Islands); 78 (Bering Sea Slope); 98 (Eastern Bering Sea Shelf); 143 (Northern Bering Sea Shelf). The column 'survey_definition_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)

Year the observation (survey) was collected.

7.1.4. AKFIN_CATCH

snapshot table for snapshot GAP_PRODUCTS.AKFIN_CATCH

Number of rows: 954,613

Number of columns: 6

Column name from data

Descriptive column Name

Units

Oracle data type

Column description

CATCHJOIN

Catch observation ID

ID key code

NUMBER(38,0)

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

COUNT

Taxon count

count, whole number resolution

NUMBER(38,0)

Total whole number of individuals caught in haul or samples collected.

CRUISEJOIN

Cruise ID

ID key code

NUMBER(38,0)

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

HAULJOIN

Haul ID

ID key code

NUMBER(38,0)

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

SPECIES_CODE

Taxon code

ID key 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

Sample or taxon weight (kg)

kilograms

NUMBER(38,3)

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

7.1.5. AKFIN_CPUE

This table is a copy of GAP_PRODUCTS.CPUE and does not have any other object dependencies. These data are produced by the Resource Assessment and Conservation Engineering Division (RACE) Groundfish Assessment Program (GAP) of the Alaska Fisheries Science Center (AFSC). There are legal restrictions on access to the data. These data are not intended for public dissemination and should not be shared without the explicit written consent of the data managers and owners (NOAA Fisheries). The GitHub repository for the scripts that created this code can be found at https://github.com/afsc-gap-products/gap_products. These data were last updated April 10, 2024.

Number of rows: 21,360,338

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 or samples collected.

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 key code

NUMBER(38,0)

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

SPECIES_CODE

Taxon code

ID key 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

Sample or taxon weight (kg)

kilograms

NUMBER(38,3)

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

7.1.6. AKFIN_CRUISE

This is the cruise data table. These data are produced by the Resource Assessment and Conservation Engineering Division (RACE) Groundfish Assessment Program (GAP) of the Alaska Fisheries Science Center (AFSC). There are legal restrictions on access to the data. These data are not intended for public dissemination and should not be shared without the explicit written consent of the data managers and owners (NOAA Fisheries). The GitHub repository for the scripts that created this code can be found at https://github.com/afsc-gap-products/gap_products. These data were last updated April 10, 2024.

Number of rows: 170

Number of columns: 10

Column name from data

Descriptive column Name

Units

Oracle data type

Column description

CRUISE

Cruise Name

ID key 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 key code

NUMBER(38,0)

Unique integer 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

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 key code

NUMBER(38,0)

The survey definition ID key code uniquely identifies a survey/survey design. Integer code that uniquely identifies survey. Full list of survey definition IDs are in RACE_-DATA.SURVEY_DEFINITIONS. IDs used in GAP_PRODUCTS are: 47 (Gulf of Alaska); 52 (Aleutian Islands); 78 (Bering Sea Slope); 98 (Eastern Bering Sea Shelf); 143 (Northern Bering Sea Shelf). The column 'survey_definition_id' is associated with the 'srvy' and 'survey' columns. For a complete list of surveys, review the code books.

SURVEY_NAME

Survey name official

text

VARCHAR2(255 BYTE)

Long name of the survey conducted

VESSEL_ID

Vessel ID

ID key 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 key 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 key codes, review the code books.

YEAR

Survey year

year

NUMBER(10,0)

Year the observation (survey) was collected.

7.1.7. AKFIN_HAUL

snapshot table for snapshot GAP_PRODUCTS.AKFIN_HAUL

Number of rows: 33,606

Number of columns: 25

Column name from data

Descriptive column Name

Units

Oracle data type

Column description

ACCESSORIES

Type of gear accessories used on the net

ID key code

NUMBER(38,0)

Type of accessories used on net. For a complete list of accessories ID key codes, review the code books.

BOTTOM_TYPE

Seafloor bottom type code

ID key code

NUMBER(38,0)

Bottom type on sea floor at haul location. For a complete list of bottom type ID key codes, review the code books.

CRUISEJOIN

Cruise ID

ID key 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

TIMESTAMP

The date (MM/DD/YYYY) and time (HH:MM) of the beginning of the haul. All dates and times are in Alaska time (AKDT) of Anchorage, AK, USA (UTC/GMT -8 hours).

DEPTH_GEAR_M

Depth of gear (m)

degrees Celsius

NUMBER(38,1)

Depth of gear (meters).

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 (thousands 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

Type of gear used on the net

ID key code

NUMBER(38,0)

Type of gear used on net. For a complete list of gear ID key codes, review the code books.

GEAR_TEMPERATURE_C

Gear temperature (degrees Celsius)

degrees Celsius

NUMBER(38,1)

Temperature recorded by net gear (tenths of a degree Celsius); NA indicates removed or missing values.

HAUL

Haul number

ID key 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 key code

NUMBER(38,0)

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

HAUL_TYPE

Haul sampling type

ID key code

NUMBER(38,0)

Type of haul sampling method. For a complete list of haul type ID key 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.

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

BINARY_DOUBLE

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 key code

VARCHAR2(255 BYTE)

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

STRATUM

Stratum ID

ID key 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.

7.1.8. AKFIN_LENGTH

snapshot table for snapshot GAP_PRODUCTS.AKFIN_LENGTH

Number of rows: 4,373,452

Number of columns: 7

Column name from data

Descriptive column Name

Units

Oracle data type

Column description

FREQUENCY

Count of observation

count

NUMBER(38,0)

Frequency, or count, of an observation.

HAULJOIN

Haul ID

ID key 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

Length type

ID key code

NUMBER(38,0)

How the taxon was measured (e.g., fork length, carapace width). For a complete list of length_type ID key codes, review the code books.

SAMPLE_TYPE

Sample type

ID key code

NUMBER(38,0)

Sampling information on how the taxon was sampled. For a complete list of length_-type ID key codes, review the code books.

SEX

Sex of a specimen

ID key code

NUMBER(38,0)

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

SPECIES_CODE

Taxon code

ID key code

NUMBER(38,0)

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

7.1.9. AKFIN_METADATA_COLUMN

This table is a copy of GAP_PRODUCTS.METADATA_COLUMN and does not have any other object dependencies. These data are produced by the Resource Assessment and Conservation Engineering Division (RACE) Groundfish Assessment Program (GAP) of the Alaska Fisheries Science Center (AFSC). There are legal restrictions on access to the data. These data are not intended for public dissemination and should not be shared without the explicit written consent of the data managers and owners (NOAA Fisheries). The GitHub repository for the scripts that created this code can be found at https://github.com/afsc-gap-products/gap_products. These data were last updated April 10, 2024.

Number of rows: 161

Number of columns: 5

Column name from data

Descriptive column Name

Units

Oracle data type

Column description

METADATA_COLNAME

Column name

text

VARCHAR2(4000 BYTE)

Name of the column in a table.

METADATA_COLNAME_DESC

Column description

text

VARCHAR2(4000 BYTE)

Description of the column.

METADATA_COLNAME_LONG

Column name spelled out

text

VARCHAR2(4000 BYTE)

Long name for the column.

METADATA_DATATYPE

Oracle datatype code

text

VARCHAR2(4000 BYTE)

Oracle data type of data column.

METADATA_UNITS

Units

category

VARCHAR2(4000 BYTE)

Units of the column.

7.1.10. AKFIN_SIZECOMP

This table is a copy of GAP_PRODUCTS.SIZECOMP and does not have any other object dependencies. These data are produced by the Resource Assessment and Conservation Engineering Division (RACE) Groundfish Assessment Program (GAP) of the Alaska Fisheries Science Center (AFSC). There are legal restrictions on access to the data. These data are not intended for public dissemination and should not be shared without the explicit written consent of the data managers and owners (NOAA Fisheries). The GitHub repository for the scripts that created this code can be found at https://github.com/afsc-gap-products/gap_products. These data were last updated April 10, 2024.

Number of rows: 3,209,517

Number of columns: 7

Column name from data

Descriptive column Name

Units

Oracle data type

Column description

AREA_ID

Area ID

ID key code

NUMBER(38,0)

Area ID key 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(38,0)

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

SEX

Sex of a specimen

ID key code

NUMBER(38,0)

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

SPECIES_CODE

Taxon code

ID key 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 key code

NUMBER(38,0)

The survey definition ID key code uniquely identifies a survey/survey design. Integer code that uniquely identifies survey. Full list of survey definition IDs are in RACE_-DATA.SURVEY_DEFINITIONS. IDs used in GAP_PRODUCTS are: 47 (Gulf of Alaska); 52 (Aleutian Islands); 78 (Bering Sea Slope); 98 (Eastern Bering Sea Shelf); 143 (Northern Bering Sea Shelf). The column 'survey_definition_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)

Year the observation (survey) was collected.

7.1.11. AKFIN_SPECIMEN

snapshot table for snapshot GAP_PRODUCTS.AKFIN_SPECIMEN

Number of rows: 576,760

Number of columns: 12

Column name from data

Descriptive column Name

Units

Oracle data type

Column description

AGE

Taxon age bin (yrs)

integer

NUMBER(38,0)

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

AGE_DETERMINATION_METHOD

Aging method

ID key code

NUMBER(10,0)

Numeric code corresponding to the method of age determination. For a complete list of age determination codes, review the code books.

GONAD_G

Weight of gonads (g)

grams

NUMBER(38,1)

Weight of specimen gonads (grams).

HAULJOIN

Haul ID

ID key 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

Specimen maturity code

ID key code

NUMBER(38,0)

The maturity code or the condition identified by the maturity code.

SEX

Sex of a specimen

ID key code

NUMBER(38,0)

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

SPECIES_CODE

Taxon code

ID key 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

Specimen unique ID

ID key code

NUMBER(38,0)

Each individual examined must have a number assigned to it that is unique within each haul (0001 to 9999), though specimen numbers may be repeated between hauls

SPECIMEN_SAMPLE_TYPE

Specimen sample type

ID key code

NUMBER(38,0)

The specimen sample type ID key code as defined in the RACE_DATA.SPECIMEN_SAMPLE_TYPES table. For a complete list of Specimen sample type ID key codes, review the code books.

SPECIMEN_SUBSAMPLE_METHOD

Specimen subsample method

ID key code

NUMBER(38,0)

For a complete list of specimen subsample method ID key codes, review the code books.

WEIGHT_G

Specimen weight (g)

grams

NUMBER(38,1)

Weight of specimen (grams).

7.1.12. AKFIN_STRATUM_GROUPS

This table is a copy of GAP_PRODUCTS.STRATUM_GROUPS and does not have any other object dependencies. These data are produced by the Resource Assessment and Conservation Engineering Division (RACE) Groundfish Assessment Program (GAP) of the Alaska Fisheries Science Center (AFSC). There are legal restrictions on access to the data. These data are not intended for public dissemination and should not be shared without the explicit written consent of the data managers and owners (NOAA Fisheries). The GitHub repository for the scripts that created this code can be found at https://github.com/afsc-gap-products/gap_products. These data were last updated April 10, 2024.

Number of rows: 768

Number of columns: 4

Column name from data

Descriptive column Name

Units

Oracle data type

Column description

AREA_ID

Area ID

ID key code

NUMBER(38,0)

Area ID key 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 key 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 key code

NUMBER(38,0)

The survey definition ID key code uniquely identifies a survey/survey design. Integer code that uniquely identifies survey. Full list of survey definition IDs are in RACE_-DATA.SURVEY_DEFINITIONS. IDs used in GAP_PRODUCTS are: 47 (Gulf of Alaska); 52 (Aleutian Islands); 78 (Bering Sea Slope); 98 (Eastern Bering Sea Shelf); 143 (Northern Bering Sea Shelf). The column 'survey_definition_id' is associated with the 'srvy' and 'survey' columns. For a complete list of surveys, review the code books.

7.1.13. AKFIN_SURVEY_DESIGN

This table is a copy of GAP_PRODUCTS.SURVEY_DESIGN and does not have any other object dependencies. These data are produced by the Resource Assessment and Conservation Engineering Division (RACE) Groundfish Assessment Program (GAP) of the Alaska Fisheries Science Center (AFSC). There are legal restrictions on access to the data. These data are not intended for public dissemination and should not be shared without the explicit written consent of the data managers and owners (NOAA Fisheries). The GitHub repository for the scripts that created this code can be found at https://github.com/afsc-gap-products/gap_products. These data were last updated April 10, 2024.

Number of rows: 87

Number of columns: 3

Column name from data

Descriptive column Name

Units

Oracle data type

Column description

DESIGN_YEAR

Design year

year

NUMBER(10,0)

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 key code

NUMBER(38,0)

The survey definition ID key code uniquely identifies a survey/survey design. Integer code that uniquely identifies survey. Full list of survey definition IDs are in RACE_-DATA.SURVEY_DEFINITIONS. IDs used in GAP_PRODUCTS are: 47 (Gulf of Alaska); 52 (Aleutian Islands); 78 (Bering Sea Slope); 98 (Eastern Bering Sea Shelf); 143 (Northern

Bering Sea Shelf). The column 'survey_definition_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)

Year the observation (survey) was collected.

7.1.14. AKFIN_TAXONOMIC_CLASSIFICATION

NAThese data are produced by the Resource Assessment and Conservation Engineering Division (RACE) Groundfish Assessment Program (GAP) of the Alaska Fisheries Science Center (AFSC). There are legal restrictions on access to the data. These data are not intended for public dissemination and should not be shared without the explicit written consent of the data managers and owners (NOAA Fisheries). The GitHub repository for the scripts that created this code can be found at https://github.com/afsc-gap-products/gap_products. These data were last updated April 10, 2024.

Number of rows: 2,699

Number of columns: 19

Column name from data

Descriptive column Name

Units

Oracle data type

Column description

CLASS_TAXON

Class phylogenetic rank

category

VARCHAR2(255 BYTE)

Phylogenetic latin rank of class 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 key code

VARCHAR2(255 BYTE)

Species ID key code of a species in the taxonomic “DATABASE” source.

FAMILY_TAXON

Family phylogenetic rank

category

VARCHAR2(255 BYTE)

Phylogenetic latin rank of family of a given species.

GENUS_TAXON

Genus phylogenetic rank

category

VARCHAR2(255 BYTE)

Phylogenetic latin rank of genus 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 of a given species.

ORDER_TAXON

Order phylogenetic rank

category

VARCHAR2(255 BYTE)

Phylogenetic latin rank of order of a given species.

PHYLUM_TAXON

Phylum phylogenetic rank

category

VARCHAR2(255 BYTE)

Phylogenetic latin rank of phylum of a given species.

SPECIES_CODE

Taxon code

ID key 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

Scientific name of species

text

VARCHAR2(255 BYTE)

Scientific name of species.

SUBCLASS_TAXON

Subclass phylogenetic rank

category

VARCHAR2(255 BYTE)

Phylogenetic latin rank of subclass of a given species.

SUBFAMILY_TAXON

Subfamily phylogenetic rank

category

VARCHAR2(255 BYTE)

Phylogenetic latin rank of subfamily of a given species.

SUBORDER_TAXON

Suborder phylogenetic rank

category

VARCHAR2(255 BYTE)

Phylogenetic latin rank of suborder of a given species.

SUBPHYLUM_TAXON

Subphylum phylogenetic rank

category

VARCHAR2(255 BYTE)

Phylogenetic latin rank of subphylum of a given species.

SUPERCLASS_TAXON

Superclass phylogenetic rank

category

VARCHAR2(255 BYTE)

Phylogenetic latin rank of superclass of a given species.

SUPERFAMILY_TAXON

Superfamily phylogenetic rank

category

VARCHAR2(255 BYTE)

Phylogenetic latin rank of superfamily of a given species.

SUPERORDER_TAXON

Superorder phylogenetic rank

category

VARCHAR2(255 BYTE)

Phylogenetic latin rank of superorder of a given species.

8. Access data via Oracle and R

Access data via Oracle (AFSC only)

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

8.0.1. 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 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.

After you connect to VPN, you'll be able to log into Oracle.

```
library(RODBC)
channel <- gapindex::get_connected()
```

Data SQL Query Examples:

```
library(gapindex)
library(RODBC)
library(flextable)
library(ggplot2)
library(magrittr)
library(dplyr)
```

8.0.1. Ex. Select all data from tables

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

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

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")))
}
```

```
library(odbc)
library(RODBC)
library(dbplyr)
```

```
my_spp_codes <- c(
  30010, #  Sebastolobus sp.    thornyhead unid.
  30020, #  Sebastolobus alascanus  shortspine thornyhead
  30025, #  Sebastolobus macrochir  broadfin thornyhead
  30330, #  Sebastes melanops   black rockfish
  30430, #  Sebastes proriger   redstripe rockfish
  30470, #  Sebastes ruberrimus yelloweye rockfish
  30475, #  Sebastes babcocki   redbanded rockfish
  30535, #  Sebastes variegatus harlequin rockfish
  30560, # Sebastes zacentrus   sharpchin rockfish
  30600, # Sebastes reedi   yellowmouth rockfish
  30030, # Sebastolobus altivelis   longspine thornyhead
  30040, # Sebastes sp. rockfish unid.
  30100, # Sebastes brevispinis silvergray rockfish
  30150, # NA   dusky and dark rockfishes unid.
  30152, # Sebastes variabilis  dusky rockfish
  30170, # Sebastes crameri darkblotched rockfish
  30270) # Sebastes helvomaculatus  rosethorn rockfish

a <- dplyr::tbl(channel, dplyr::sql('gap_products.akfin_biomass')) %>%
  dplyr::rename_all(tolower) %>%
  dplyr::select(survey_definition_id, area_id, species_code, year, biomass_mt, biomass_var)
  dplyr::filter(species_code %in% my_spp_codes &
                area_id %in% 99904 &
                year >= 1991) %>%
  dplyr::collect()

flextable::flextable(head(a)) %>%
  flextable::fit_to_width(max_width = 6) %>%
  flextable::theme_zebra()
```

8.0.2. Ex. CPUE for all EBS and NBS stations with associated haul, cruise, and species information.

```
a <- RODBC::sqlQuery(channel = channel, # NOT RACEBASE.HAUL
                     query = paste0(
                       "
-- Select columns for output data
```

```
SELECT
cr.CRUISEJOIN,
cr.CRUISE,
cr.YEAR,
cr.SURVEY_DEFINITION_ID,
cr.SURVEY_NAME,
cr.VESSEL_ID,
cr.VESSEL_NAME,
cp.HAULJOIN,
cp.SPECIES_CODE,
tt.SPECIES_NAME,
tt.COMMON_NAME,
cp.WEIGHT_KG,
cp.COUNT,
cp.AREA_SWEPT_KM2,
cp.CPUE_KGKM2,
cp.CPUE_NOKM2,
hh.HAUL,
hh.STATION

-- Identify what tables to pull data from
FROM GAP_PRODUCTS.AKFIN_HAUL hh
LEFT JOIN GAP_PRODUCTS.AKFIN_CRUISE cr
ON hh.CRUISEJOIN = cr.CRUISEJOIN
LEFT JOIN GAP_PRODUCTS.AKFIN_CPUE cp
ON hh.HAULJOIN = cp.HAULJOIN
LEFT JOIN GAP_PRODUCTS.TAXONOMIC_CLASSIFICATION tt
ON cp.SPECIES_CODE = tt.SPECIES_CODE

-- Filter for EBS and NBS observations
WHERE SURVEY_DEFINITION_ID IN (143, 98) -- 143 NBS, 98 EBS
AND tt.SURVEY_SPECIES = 1

-- Only return the first 3 rows because otherwise this would be a huge table!
FETCH FIRST 3 ROWS ONLY;"))

flextable::flextable(head(a[,2:8])) %>%
  flextable::fit_to_width(max_width = 6) %>%
  flextable::theme_zebra()
```

Table 8.1.: CPUE for all EBS and NBS stations with associated haul, cruise, and species information.

CRUISE	YEAR	SURVEY_DEFINITION_ID	SURVEY_NAME	VESSEL_ID	VESSEL_NAME	HAULJOIN
202,301	2,023	98	Eastern Bering Sea Crab/Grou Bottom Trawl Survey	162	ALASKA KNIGHT	-22,807
202,301	2,023	98	Eastern Bering Sea Crab/Groundfish Bottom Trawl Survey	134	NORTHWEST EXPLORER	-22,788
202,301	2,023	98	Eastern Bering Sea Crab/Grou Bottom Trawl Survey	134	NORTHWES EXPLORER	-22,755

8.0.3. Ex. CPUE for all stations contained in the INPFC Shumagin region (AREA_ID = 919) for Pacific cod.

```
dat <- RODBC::sqlQuery(channel = channel,
                       query =
                         "
-- Select columns for output data
SELECT
HAULJOIN,
SPECIES_CODE,
STRATUM,
LATITUDE_DD_START,
```

```
LONGITUDE_DD_START,
CPUE_KGKM2,
GEAR_TEMPERATURE_C

-- Identify what tables to pull data from
FROM GAP_PRODUCTS.AKFIN_CPUE cpue
LEFT JOIN GAP_PRODUCTS.AKFIN_HAUL haul
USING (HAULJOIN)

-- Filter for P. Cod observations
WHERE SPECIES_CODE IN (21720)

-- Select all stratum within the area_id 919 (INPFC Shumagin region)
AND haul.STRATUM IN
(
SELECT
STRATUM
FROM GAP_PRODUCTS.AKFIN_STRATUM_GROUPS
WHERE AREA_ID = 919
);")
```

```
dat <- dat %>%
  dplyr::select(HAULJOIN, STRATUM, SPECIES_CODE, LATITUDE_DD_START, LONGITUDE_DD_START, CPU
  dplyr::mutate(SPECIES_CODE = as.character(SPECIES_CODE),
                STRATUM = as.character(STRATUM)) %>%
  dplyr::arrange(SPECIES_CODE)

flextable::flextable(head(dat)) %>%
  flextable::fit_to_width(max_width = 6) %>%
  flextable::theme_zebra()
```

Table 8.2.: CPUE for all stations contained in the Shumagin region (AREA_ID = 919).

HAULJOIN	STRATUM	SPECIES_-CODE	LATITUDE_DD_-START	LONGITUD DD_-START	CPUE_-KGKM2	GEAR_-TEMPER-ATURE_C
-22,269	13	21720	54.93934	-159.6317	400.3135	5.4
-22,272	13	21720	55.10890	-159.1346	156.6961	4.1

HAULJOIN	STRATUM	SPECIES_-CODE	LATITUDE_DD_-START	LONGITUDE_DD_-START	CPUE_-KGKM2	GEAR_-TEMPER-ATURE_C
-22,271	112	21720	55.22129	-159.1538	1,263.8377	4.1
-22,270	13	21720	55.11515	-159.3512	209.2899	4.3
-22,054	111	21720	53.46000	-166.2102	1,142.3927	4.9
-22,049	111	21720	52.99132	-167.8501	3,723.9059	4.5

8.0.4. Ex. 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 columns for output data
SELECT
(cp.CPUE_KGKM2/100) CPUE_KGHA, -- akgfmaps is expecting hectares
hh.LATITUDE_DD_START LATITUDE,
hh.LONGITUDE_DD_START LONGITUDE

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

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

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

```
dat %>%
  dplyr::arrange(desc(CPUE_KGHA)) %>%
  head() %>%
  flextable::flextable() %>%
  flextable::fit_to_width(max_width = 6) %>%
  flextable::theme_zebra()
```

Table 8.3.: EBS Pacific Ocean perch CPUE and `akgfmaps` map.

CPUE_-KGHA	LATITUDE	LONGITUDE
10.1768965	57.64871	-173.3735
6.2734470	56.36952	-169.4604
3.0252034	56.66253	-171.9549
1.8214628	57.98912	-173.4816
0.5535672	55.65865	-168.1804
0.2813533	57.32545	-173.3217

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

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

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

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

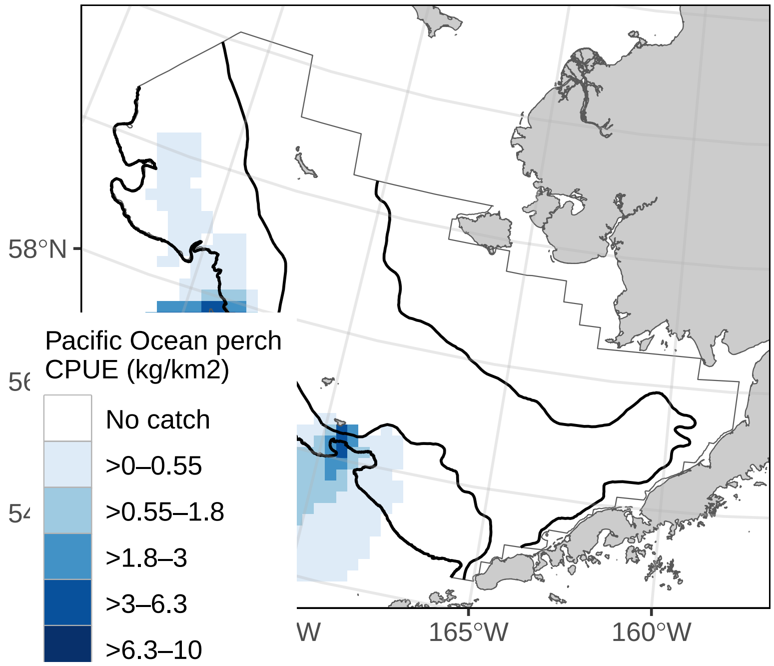


Figure 8.1.: EBS Pacific Ocean perch CPUE and `akgfmaps` map.

8.0.5. Ex. 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 =
                         "
-- Manipulate data to join to
WITH FILTERED_STRATA AS (
SELECT AREA_ID, DESCRIPTION FROM GAP_PRODUCTS.AKFIN_AREA
WHERE AREA_TYPE in ('REGULATORY_AREA', 'REGION')
AND SURVEY_DEFINITION_ID = 47)

-- Select columns for output data
SELECT
BIOMASS_MT,
POPULATION_COUNT,
```

```
YEAR,
DESCRIPTION

-- Identify what tables to pull data from
FROM GAP_PRODUCTS.AKFIN_BIOMASS BIOMASS
JOIN FILTERED_STRATA STRATA
ON STRATA.AREA_ID = BIOMASS.AREA_ID

-- Filter data results
WHERE 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)) %>%
  flextable::fit_to_width(max_width = 6) %>%
  flextable::theme_zebra() %>%
  flextable::colformat_num(x = ., j = "YEAR", big.mark = "")
```

Table 8.4.: GOA Pacific Ocean perch biomass and abundance.

BIOMASS_MT	POPULATION COUNT	YEAR	DESCRIPTION
157,295.1	317,129,40	1990	GOA Region: All Strata
157,295.1	317,129,408	1990	GOA Region: All Strata
483,622.6	833,902,16	1993	GOA Region: All Strata
483,622.6	833,902,161	1993	GOA Region: All Strata
771,412.8	1,252,616,6	1996	GOA Region: All Strata
771,412.8	1,252,616,603	1996	GOA Region: All Strata

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

```
                 legend.position = "bottom")

figure
```

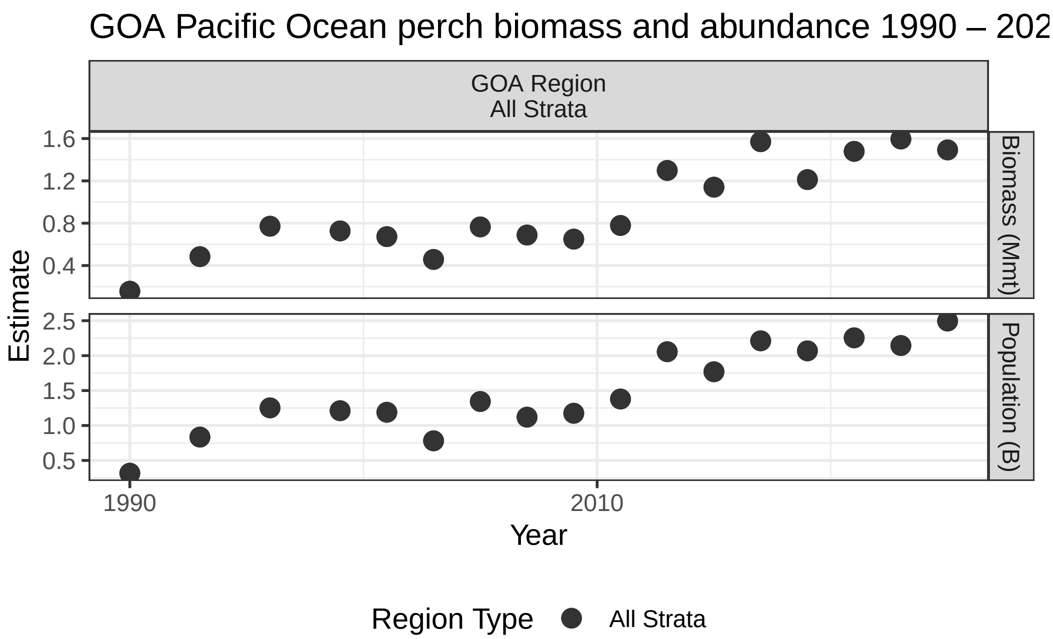


Figure 8.2.: GOA Pacific Ocean perch biomass and abundance.

8.0.6. Ex. 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 = "
SELECT
YEAR,
LENGTH_MM / 10 AS LENGTH_CM,
SUM(POPULATION_COUNT) AS POPULATION_COUNT

-- Identify what tables to pull data from
```

```
FROM GAP_PRODUCTS.AKFIN_SIZECOMP

-- 99904 is the AREA_ID that codes for the whole AI survey region
WHERE AREA_ID = 99904
-- including northern rock sole, southern rock sole, and rock sole unid.
AND SPECIES_CODE IN (10260, 10261, 10262)
-- remove the -9 LENGTH_MM code
AND LENGTH_MM > 0
-- sum over species_codes and sexes
GROUP BY (YEAR, LENGTH_MM)")
```

```
dat0 <- dat %>%
  janitor::clean_names() %>%
  head() %>%
  flextable::flextable() %>%
  flextable::fit_to_width(max_width = 6) %>%
  flextable::theme_zebra() %>%
  flextable::colformat_num(x = ., j = "year", big.mark = "")
dat0
```

Table 8.5.: AI Rock sole size compositions and ridge plot.

year	length_-cm	population_-count
1991	23	4,625,236
1991	38	2,254,964
1991	42	820,614
1991	52	11,225
1994	16	741,246
1994	26	9,762,322

```
# install.packages("ggridges")
library(ggridges)
figure <- ggplot(dat,
                 mapping = aes(x = LENGTH_CM,
                               y = YEAR,
```

```
                                height = POPULATION_COUNT,
                                group = YEAR)) +
  ggridges::geom_density_ridges(stat = "identity", scale = 1) +
  ggplot2::ylab(label = "Year") +
  ggplot2::scale_x_continuous(name = "Length (cm)") +
  ggplot2::labs(title = paste0('Aleutian Islands Rock sole Size Compositions'),
                subtitle = paste0(min(dat$YEAR), ' - ', max(dat$YEAR))) +
  ggplot2::theme_bw()

figure
```

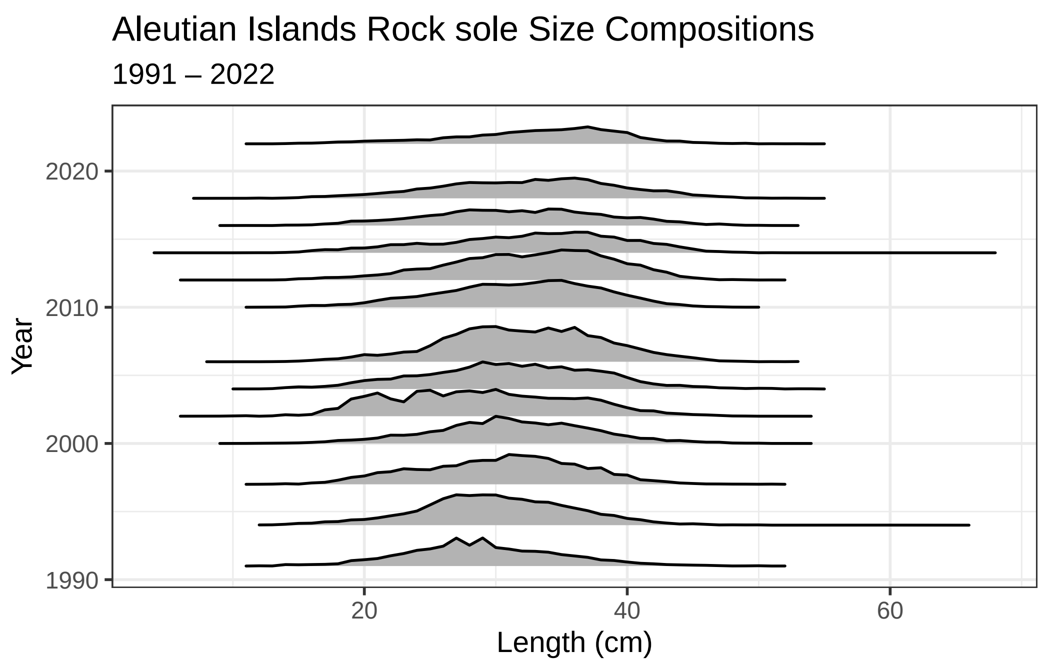


Figure 8.3.: AI Rock sole size compositions and ridge plot.

8.0.7. Ex. 2023 EBS Walleye Pollock Age Compositions and Age Pyramid

Walleye pollock age composition for the EBS standard + NW Area from 2023, with age pyramid plot.

```
dat <- RODBC::sqlQuery(channel = channel,
                       query = "
-- Manipulate data to join to
WITH FILTERED_STRATA AS (
SELECT
AREA_ID,
DESCRIPTION
FROM GAP_PRODUCTS.AKFIN_AREA
WHERE AREA_TYPE = 'REGION' AND
SURVEY_DEFINITION_ID = 98)

-- Select columns for output data
SELECT
AGECOMP.AGE,
AGECOMP.POPULATION_COUNT,
AGECOMP.SEX

-- Identify what tables to pull data from
FROM GAP_PRODUCTS.AKFIN_AGECOMP AGECOMP
JOIN FILTERED_STRATA STRATA
ON STRATA.AREA_ID = AGECOMP.AREA_ID

-- Filter data results
WHERE SPECIES_CODE = 21740
AND YEAR = 2023
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)) %>%
  flextable::fit_to_width(max_width = 6) %>%
  flextable::theme_zebra()
```

Table 8.6.: EBS Walleye Pollock Age Compositions and Age Pyramid.

AGE	POPULATI COUNT	SEX
1	22,107,475	1
2	123,357,791	1
3	136,795,03	1
4	253,072,637	1
5	967,413,01	1
6	242,965,335	1

```
figure <- ggplot2::ggplot(
  data = dat0,
  mapping =
    aes(x = age,
        y = population_count,
        fill = sex)) +
  ggplot2::scale_fill_grey() +
  ggplot2::geom_bar(stat = "identity") +
  ggplot2::coord_flip() +
  ggplot2::scale_x_continuous(name = "Age") +
  ggplot2::scale_y_continuous(name = "Population (billions)", labels = abs) +
  ggplot2::ggtitle(label = "2023 EBS Walleye Pollock Age Compositions")  +
  ggplot2::guides(fill = guide_legend(title = "Sex"))+
  ggplot2::theme_bw()

figure
```

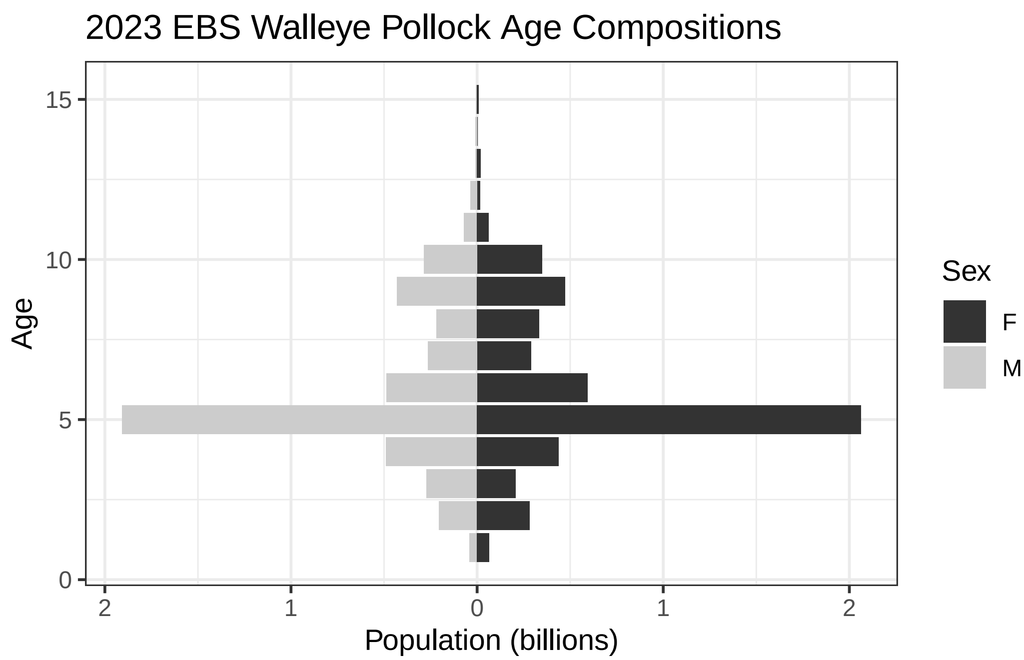


Figure 8.4.: 2023 EBS Walleye Pollock Age Compositions and Age Pyramid.

8.0.8. Ex. NBS Pacific cod biomass and abundance

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

```
dat <- RODBC::sqlQuery(channel = channel,
                       query =
                         "
-- Manipulate data to join to
WITH FILTERED_STRATA AS (
SELECT
AREA_ID,
AREA_NAME,
DESCRIPTION
FROM GAP_PRODUCTS.AKFIN_AREA
WHERE AREA_TYPE in ('STRATUM') AND
SURVEY_DEFINITION_ID = 143)

-- Select columns for output data
```

```
SELECT
BIOMASS.BIOMASS_MT,
BIOMASS.POPULATION_COUNT,
BIOMASS.YEAR,
STRATA.AREA_NAME

-- Identify what tables to pull data from
FROM GAP_PRODUCTS.AKFIN_BIOMASS BIOMASS
JOIN FILTERED_STRATA STRATA
ON STRATA.AREA_ID = BIOMASS.AREA_ID

-- Filter data results
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)) %>%
  flextable::fit_to_width(max_width = 6) %>%
  flextable::theme_zebra() %>%
  flextable::colformat_num(x = ., j = "YEAR", big.mark = "")
```

Table 8.7.: NBS Pacific cod biomass and abundance.

BIOMASS_-MT	POPULATI COUNT	YEAR	AREA_-NAME
7,462.559	4,724,153	2010	Inner Domain
7,462.559	4,724,153	2010	Inner Domain

Data SQL Query Examples:

BIOMASS_MT	POPULATION_COUNT	YEAR	AREA_NAME
7,462.559	4,724,153	2010	Inner Domain
7,462.559	4,724,153	2010	Inner Domain
7,462.559	4,724,153	2010	Inner Domain
20,983.376	3,928,600	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
```

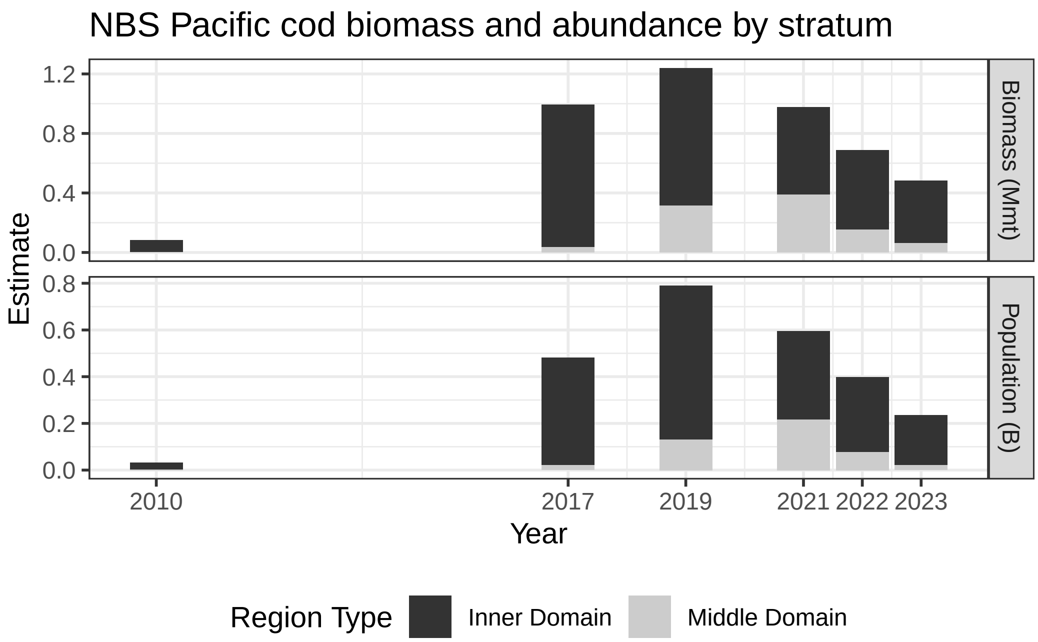


Figure 8.5.: NBS Pacific cod biomass and abundance.

8.0.9. Ex. 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 columns for output data
SELECT
SURVEY_DEFINITION_ID,
BIOMASS_MT / 1000 AS BIOMASS_KMT,
(BIOMASS_MT - 2 * SQRT(BIOMASS_VAR)) / 1000 AS BIOMASS_KCI_DW,
(BIOMASS_MT + 2 * SQRT(BIOMASS_VAR)) / 1000 AS BIOMASS_KCI_UP,
YEAR

-- Identify what tables to pull data from
FROM GAP_PRODUCTS.AKFIN_BIOMASS
```

```
-- Filter data results
WHERE SPECIES_CODE = 30060
AND SURVEY_DEFINITION_ID = 47
AND AREA_ID = 99903
AND YEAR BETWEEN 1990 AND 2023" ) %>%
  janitor::clean_names()
```

```
flextable::flextable(head(dat)) %>%
  flextable::fit_to_width(max_width = 6) %>%
  flextable::theme_zebra() %>%
  flextable::colformat_num(x = ., j = "year", big.mark = "")
```

Table 8.8.: GOA Pacific Ocean perch biomass and line plot.

survey_-definition_id	biomass_-kmt	biomass_-kci_dw	biomass_-kci_up	year
47	157.2951	63.03638	251.5538	1990
47	483.6226	266.33581	700.9093	1993
47	771.4128	364.30515	1,178.5204	1996
47	764.9014	471.64517	1,058.1577	2005
47	688.1798	459.83542	916.5241	2007
47	457.4216	313.39204	601.4511	2003

```
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,
      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::ggtitle(
    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
```

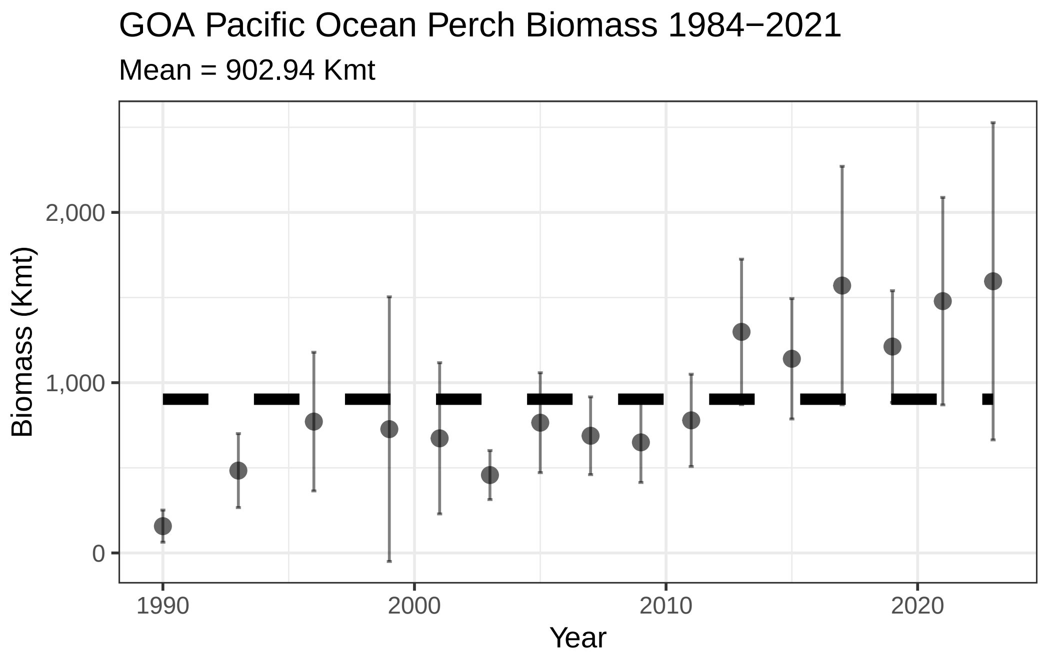


Figure 8.6.: GOA Pacific Ocean perch biomass and line plot.

8.0.10. Ex. 2022 AI Atka mackerel age specimen summary

8.0.10.1. All ages determined:

```
dat <- RODBC::sqlQuery(channel = channel,
                       query = "
-- Select columns for output data
SELECT SURVEY_DEFINITION_ID, YEAR, SPECIES_CODE, AGE

-- Identify what tables to pull data from
FROM GAP_PRODUCTS.AKFIN_SPECIMEN
JOIN (SELECT HAULJOIN, CRUISEJOIN FROM GAP_PRODUCTS.AKFIN_HAUL)
USING (HAULJOIN)
JOIN (SELECT CRUISEJOIN, YEAR, SURVEY_DEFINITION_ID FROM GAP_PRODUCTS.AKFIN_CRUISE)
USING (CRUISEJOIN)

-- Filter data results
```

```
WHERE GAP_PRODUCTS.AKFIN_SPECIMEN.SPECIMEN_SAMPLE_TYPE = 1
AND SPECIES_CODE = 21921
AND YEAR = 2022
AND SURVEY_DEFINITION_ID = 52") %>%
  janitor::clean_names()
```

```
flextable::flextable(head(dat) %>%
                       dplyr::arrange(age)) %>%
  flextable::fit_to_width(max_width = 6) %>%
  flextable::theme_zebra() %>%
  flextable::colformat_num(x = ., j = c("year", "species_code"), big.mark = "")
```

Table 8.9.: 2022 AI Atka mackerel age specimen summary: all ages determined.

survey_definition_id	year	species_code	age
52	2022	21921	3
52	2022	21921	3
52	2022	21921	4
52	2022	21921	4
52	2022	21921	4
52	2022	21921	7

8.0.10.2. How many of each age was found:

```
dat <- RODBC::sqlQuery(channel = channel,
                       query = "
-- Select columns for output data
SELECT SURVEY_DEFINITION_ID, YEAR, SPECIES_CODE, AGE,
COUNT(AGE) AS COUNTAGE

-- Identify what tables to pull data from
FROM GAP_PRODUCTS.AKFIN_SPECIMEN
```

```
JOIN (SELECT HAULJOIN, CRUISEJOIN FROM GAP_PRODUCTS.AKFIN_HAUL)
USING (HAULJOIN)
JOIN (SELECT CRUISEJOIN, YEAR, SURVEY_DEFINITION_ID FROM GAP_PRODUCTS.AKFIN_CRUISE)
USING (CRUISEJOIN)

-- Filter data results
WHERE GAP_PRODUCTS.AKFIN_SPECIMEN.SPECIMEN_SAMPLE_TYPE = 1
AND SPECIES_CODE = 21921
AND YEAR = 2022
AND SURVEY_DEFINITION_ID = 52
GROUP BY (YEAR, SURVEY_DEFINITION_ID, SPECIES_CODE, AGE)") %>%
  janitor::clean_names()
```

```
flextable::flextable(head(dat) %>%
                       dplyr::arrange(age)) %>%
  flextable::fit_to_width(max_width = 6) %>%
  flextable::theme_zebra() %>%
  flextable::colformat_num(x = ., j = c("year", "species_code"), big.mark = "")
```

Table 8.10.: Ex.: 2022 AI Atka mackerel age specimen summary: how many of each age were determined.

survey_-defini-tion_id	year	species_-code	age	countage
52	2022	21921	1	1
52	2022	21921	2	40
52	2022	21921	6	116
52	2022	21921	7	108
52	2022	21921	8	61
52	2022	21921	11	20

8.0.10.3. How many otoliths were aged:

Using `SQL`

```
dat <- RODBC::sqlQuery(channel = channel,
                       query = "
-- Select columns for output data
SELECT SURVEY_DEFINITION_ID, YEAR, SPECIES_CODE,
COUNT(AGE) AS COUNTAGE

-- Identify what tables to pull data from
FROM GAP_PRODUCTS.AKFIN_SPECIMEN
JOIN (SELECT HAULJOIN, CRUISEJOIN FROM GAP_PRODUCTS.AKFIN_HAUL)
USING (HAULJOIN)
JOIN (SELECT CRUISEJOIN, YEAR, SURVEY_DEFINITION_ID FROM GAP_PRODUCTS.AKFIN_CRUISE)
USING (CRUISEJOIN)

-- Filter data results
WHERE GAP_PRODUCTS.AKFIN_SPECIMEN.SPECIMEN_SAMPLE_TYPE = 1
AND SPECIES_CODE = 21921
AND YEAR = 2022
AND SURVEY_DEFINITION_ID = 52
GROUP BY (YEAR, SURVEY_DEFINITION_ID, SPECIES_CODE)") %>%
  janitor::clean_names()
```

Using `dbplyr`:

```
library(odbc)
library(keyring)
library(dplyr)
library(dbplyr)

channel <- DBI::dbConnect(odbc::odbc(), "akfin", uid = keyring::key_list("akfin")$username,
                        pwd = keyring::key_get("akfin", keyring::key_list("akfin")$username

dat <- dplyr::tbl(src = channel, dplyr::sql('gap_products.akfin_specimen')) %>%
    dplyr::rename_all(tolower) %>%
    dplyr::select(hauljoin, specimen = specimen_id, species_code, length = length_mm,
                  weight = weight_g, age, sex, age_method = age_determination_method) %>%
    dplyr::left_join(dplyr::tbl(akfin, dplyr::sql('gap_products.akfin_haul')) %>%
                       dplyr::rename_all(tolower) %>%
                       dplyr::select(cruisejoin, hauljoin, haul, date_collected = date_time
                                     latitude = latitude_dd_start, longitude = longitude_dd
                     by = join_by(hauljoin)) %>%
```

```
    dplyr::left_join(dplyr::tbl(akfin, dplyr::sql('gap_products.akfin_cruise')) %>%
                       dplyr::rename_all(tolower) %>%
                       dplyr::select(cruisejoin, year, vessel = vessel_id, survey_definitio
                     by = join_by(cruisejoin)) %>%
    dplyr::filter(year == YEAR &
             survey_definition_id == 52 &
             species_code %in% spp_codes &
             !is.na(age)) %>%
    dplyr::collect()
```

Both scripts will produce this table:

```
flextable::flextable(head(dat)) %>%
  flextable::fit_to_width(max_width = 6) %>%
  flextable::theme_zebra() %>%
  flextable::colformat_num(x = ., j = c("year", "species_code"), big.mark = "")
```

Table 8.11.: 2022 AI Atka mackerel age specimen summary: how many otoliths were aged. This quiery was created using SQL.

survey_-defini-tion_id	year	species_-code	countage
52	2022	21921	1,061

9. Access API data using R

AKFIN has developed web services (apis) to distribute GAP data. Like the GAP_PRODUCTS schema, these are under active development. These do not require VPN or an oracle connection but they are protected by Oracle authentication, please contact matt.callahan@noaa.gov for information on how to get an api token to use this option.

The url structure is "https://apex.psmfc.org/akfin/data_marts/gap_products/gap_[base table name]" . For example "https://apex.psmfc.org/akfin/data_marts/gap_products/gap_biomass" is the base url to get data from the akfin_biomass table. Web services linked to large tables have mandatory parameters to reduce data download size. For example to get agecomp data for Bering Sea pollock in area_id 10 in 2022 you would use "https://apex.psmfc.org/akfin/data_marts/gap_products/gap_biomass?survey_definition_id=98&area_id=10&species_code=21740&start_year=2022&end_year=2022".

If you're using R to pull data through web services you might find the akfingapdata (pronounced akfin-gap-data not ak-eff-ing-app-data) R package helpful.

```
# 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()
}
```

9.1. Ex. Direct database query in R using the (akfingapdata readme)[https://github.com/MattCallahan-NOAA/akfingapdata/blob/main/README.Rmd] R package:

```
# load packages
library(odbc)
library(getPass)
library(tidyverse)

# connect to AKFIN Oracle database
con <- dbConnect(odbc::odbc(), "akfin", UID=getPass(msg="USER NAME"), PWD=getPass())
```

```
# define species code for pollock
my_species <- 21740

#query database
data<- dbFetch(dbSendQuery(con,
                           paste0("select * from gap_products.akfin_biomass
where species_name = ", my_species,
" and survey_definition_id = 98,
and area_id = 10"))) %>%
rename_with(tolower) # everyone likes lower case letters better

head(data)
```

9.2. Ex. Direct database query in R using the (akfingapdata readme)[https://github.com/MattCallahan-NOAA/akfingapdata/blob/main/README.Rmd] R package:

```
library(akfingapdata)

# Sign into akfin with token (need to request token from AKFIN)
token <- akfingapdata::create_token(file = paste0(dirname(here::here()), "/akfin_token.txt"

akfingapdata::get_gap_catch()[,1:6] %>%
  head() %>%
  flextable::flextable() %>%
  flextable::theme_zebra()
```

Table 9.1.: Ex. 2: Load catch data with {akfingapdata}.

cruisejoin	hauljoin	catchjoin	species_-code	weight_-kg	count
-611	-13,626	-374,397	21,740	189.590	450
-611	-13,632	-374,643	21,740	134.830	151
-611	-13,501	-370,283	21,740	32.060	44
-611	-13,545	-371,701	21,740	1,154.024	1,345
-611	-13,577	-372,795	21,740	963.070	1,273
-611	-13,677	-376,146	21,740	1,618.589	2,814

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

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.*

Access Constraints

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. Acknowledgment 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.

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

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

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, 2024). Add "note = {Accessed: mm/dd/yyyy}" to append the day this data was accessed.

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

10. 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.

10.1. Data tables

10.1.1. FOSS_CATCH

These datasets, FOSS_CATCH, FOSS_CPUE_PRESONLY, FOSS_HAUL, and FOSS_-SPECIES, when full joined by the HAULJOIN variable, includes zero-filled (presence and absence) observations and catch-per-unit-effort (CPUE) estimates for all identified species at for index stations. These tables were created by the Resource Assessment and Conservation Engineering Division (RACE) Groundfish Assessment Program (GAP) of the Alaska Fisheries Science Center (AFSC). There are legal restrictions on access to the data. These data are not intended for public dissemination and should not be shared without the explicit written consent of the data managers and owners (NOAA Fisheries). The GitHub repository for the scripts that created this code can be found at https://github.com/afsc-gap-products/gap_products. For more information about codes used in the tables, please refer to the survey code books (https://www.fisheries.noaa.gov/resource/document/groundfish-survey-species-code-manual-and-data-codes-manual). These data were last updated March 04, 2024.

Number of rows: 939,197

Number of columns: 7

Column name from data

Descriptive column Name

Units

Oracle data type

Column description

COUNT

Taxon count

count, whole number resolution

NUMBER(38,0)

Total whole number of individuals caught in haul or samples collected.

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 key code

NUMBER(38,0)

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

SPECIES_CODE

Taxon code

ID key 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

Sample or taxon weight (kg)

kilograms

NUMBER(38,3)

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

10.1.2. FOSS_HAUL

These datasets, FOSS_CATCH, FOSS_CPUE_PRESONLY, FOSS_HAUL, and FOSS_-SPECIES, when full joined by the HAULJOIN variable, includes zero-filled (presence and absence) observations and catch-per-unit-effort (CPUE) estimates for all identified species at for index stations. These tables were created by the Resource Assessment and Conservation Engineering Division (RACE) Groundfish Assessment Program (GAP) of the Alaska Fisheries Science Center (AFSC). There are legal restrictions on access to the data. These data are not intended for public dissemination and should not be shared without the explicit written consent of the data managers and owners (NOAA Fisheries). The GitHub repository for the scripts that created this code can be found at https://github.com/afsc-gap-products/gap_products. For more information about codes used in the tables, please refer to the survey code books (https://www.fisheries.noaa.gov/resource/document/groundfish-survey-species-code-manual-and-data-codes-manual). These data were last updated March 04, 2024.

Number of rows: 33,334

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 Name

ID key 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 key code

NUMBER(38,0)

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. All dates and times are in Alaska time (AKDT) of Anchorage, AK, USA (UTC/GMT -8 hours).

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 (thousands 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 key 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 key code

NUMBER(38,0)

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

LATITUDE_DD_END

End latitude (decimal degrees)

decimal degrees

NUMBER(38,6)

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

LATITUDE_DD_START

Start latitude (decimal degrees)

decimal degrees

NUMBER(38,6)

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

LONGITUDE_DD_END

End longitude (decimal degrees)

decimal degrees

NUMBER(38,6)

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

LONGITUDE_DD_START

Start longitude (decimal degrees)

decimal degrees

NUMBER(38,6)

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

NET_HEIGHT_M

Net height (m)

meters

NUMBER(38,1)

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

NET_WIDTH_M

Net width (m)

meters

NUMBER(38,1)

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

PERFORMANCE

Haul performance code

category

NUMBER(38,0)

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

SRVY

Survey abbreviation

text abbreviated

VARCHAR2(255 BYTE)

Abbreviated survey names. The column 'srvy' is associated with the 'survey' and 'survey_definition_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 key code

VARCHAR2(255 BYTE)

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

STRATUM

Stratum ID

ID key 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_definition_id' columns.

SURVEY_DEFINITION_ID

Survey ID

ID key code

NUMBER(38,0)

The survey definition ID key code uniquely identifies a survey/survey design. Integer code that uniquely identifies survey. Full list of survey definition IDs are in RACE_-DATA.SURVEY_DEFINITIONS. IDs used in GAP_PRODUCTS are: 47 (Gulf of Alaska); 52 (Aleutian Islands); 78 (Bering Sea Slope); 98 (Eastern Bering Sea Shelf); 143 (Northern Bering Sea Shelf). The column 'survey_definition_id' is associated with the 'srvy' and 'survey' columns. For a complete list of surveys, review the code books.

SURVEY_NAME

Survey name official

text

VARCHAR2(255 BYTE)

Long name of the survey conducted

VESSEL_ID

Vessel ID

ID key 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 key 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 key codes, review the code books.

YEAR

Survey year

year

NUMBER(10,0)

Year the observation (survey) was collected.

10.1.3. FOSS_SPECIES

These datasets, FOSS_CATCH, FOSS_CPUE_PRESONLY, FOSS_HAUL, and FOSS_SPECIES, when full joined by the HAULJOIN variable, includes zero-filled (presence and absence) observations and catch-per-unit-effort (CPUE) estimates for all identified species at for index stations. These tables were created by the Resource Assessment and Conservation Engineering Division (RACE) Groundfish Assessment Program (GAP) of the Alaska Fisheries Science Center (AFSC). There are legal restrictions on access to the data. These data are not intended for public dissemination and should not be shared without the explicit written consent of the data managers and owners (NOAA Fisheries). The GitHub repository for the scripts that created this code can be found at https://github.com/afsc-gap-products/gap_products. For more information about codes used in the tables, please refer to the survey code books (https://www.fisheries.noaa.gov/resource/document/groundfish-survey-species-code-manual-and-data-codes-manual). These data were last updated March 04, 2024.

Number of rows: 1,894

Number of columns: 6

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.

ID_RANK

Lowest taxonomic rank

text

VARCHAR2(255 BYTE)

Lowest taxonomic rank of a given species entry.

ITIS

Integrated taxonomic information system (ITIS) serial number

ID key code

NUMBER(38,0)

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

SCIENTIFIC_NAME

Taxon scientific name

text

VARCHAR2(255 BYTE)

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

SPECIES_CODE

Taxon code

ID key 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.

WORMS

World register of marine species (WoRMS) taxonomic serial number

ID key code

NUMBER(38,0)

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

10.1.4. FOSS_SURVEY_SPECIES

This reference dataset contains the full list of species by survey to be used to zero-fill FOSS_CATCH and FOSS_HAUL for each survey. These tables were created by the Resource Assessment and Conservation Engineering Division (RACE) Groundfish Assessment Program (GAP) of the Alaska Fisheries Science Center (AFSC). There are legal restrictions on access to the data. These data are not intended for public dissemination and should not be shared without the explicit written consent of the data managers and owners (NOAA Fisheries). The GitHub repository for the scripts that created this code can be found at https://github.com/afsc-gap-products/gap_products. For more information about codes used in the tables, please refer to the survey code books (https://www.fisheries.noaa.gov/resource/document/groundfish-survey-species-code-manual-and-data-codes-manual). These data were last updated March 04, 2024.

Number of rows: 5,030

Number of columns: 2

Column name from data

Descriptive column Name

Units

Oracle data type

Column description

SPECIES_CODE

Taxon code

ID key 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 key code

NUMBER(38,0)

The survey definition ID key code uniquely identifies a survey/survey design. Integer code that uniquely identifies survey. Full list of survey definition IDs are in RACE_-DATA.SURVEY_DEFINITIONS. IDs used in GAP_PRODUCTS are: 47 (Gulf of Alaska); 52 (Aleutian Islands); 78 (Bering Sea Slope); 98 (Eastern Bering Sea Shelf); 143 (Northern Bering Sea Shelf). The column 'survey_definition_id' is associated with the 'srvy' and 'survey' columns. For a complete list of surveys, review the code books.

10.1.5. FOSS_TAXON_GROUP

This reference dataset contains suggested search groups for simplifying species selection in the FOSS data platform so users can better search through FOSS_-CATCH. These tables were created by the Resource Assessment and Conservation Engineering Division (RACE) Groundfish Assessment Program (GAP) of the Alaska Fisheries Science Center (AFSC). There are legal restrictions on access to the data. These data are not intended for public dissemination and should not be shared without the explicit written consent of the data managers and owners (NOAA Fisheries). The GitHub repository for the scripts that created this code can be found at https://github.com/afsc-gap-products/gap_products. For more information about codes used in the tables, please refer to the survey code books (https://www.fisheries.noaa.gov/resource/document/groundfish-survey-species-code-manual-and-data-codes-manual). These data were last updated March 04, 2024.

Number of rows: 33,721

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

Taxonomic rank

category

VARCHAR2(255 BYTE)

The taxonomic rank of a taxon identification.

SPECIES_CODE

Taxon code

ID key 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.

11. Using the FOSS platform

11.1. Select and filter

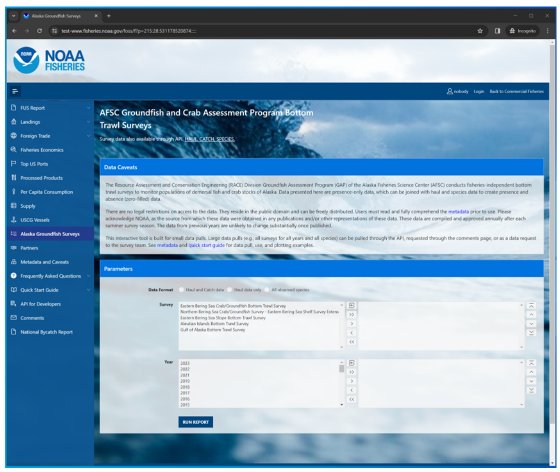


Figure 11.1.: AFSC Groundfish and Crab Assessment Program Bottom Trawl Survey data interface on the Fisheries One Stop Shop platform.

Select, filter, and download 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 the kind of data you need: `Haul and catch data`, `Haul data only`, `All observed species`.

> In this example, we'll select for 2023 eastern Bering Sea Arctic 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.

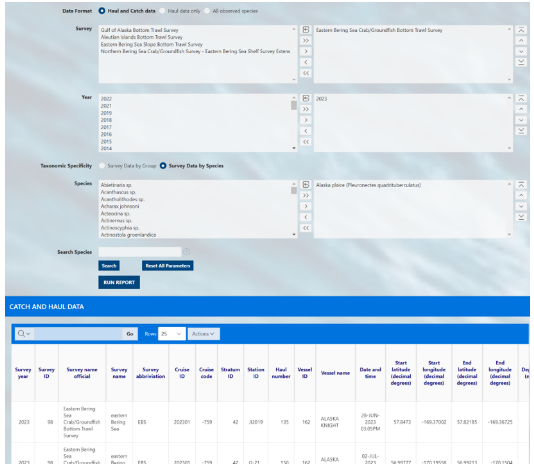


Figure 11.2.: Catch data on the AFSC Groundfish and Crab Assessment Program Bottom Trawl Survey data interface on the Fisheries One Stop Shop platform.

11.1.1. Catch and haul

11.1.2. Haul

11.1.3. Species

11.2. Search options

The user must select a option in each of the three `option boxes` as they appear for `catch`, `haul`, and `species`:

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

For a given box, select one or a few options from the `options box` (list on the left) to query. 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

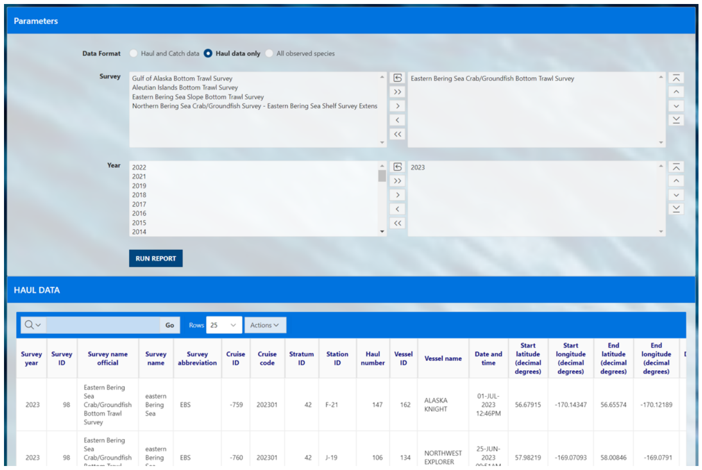


Figure 11.3.: Haul data on the AFSC Groundfish and Crab Assessment Program Bottom Trawl Survey data interface on the Fisheries One Stop Shop platform.

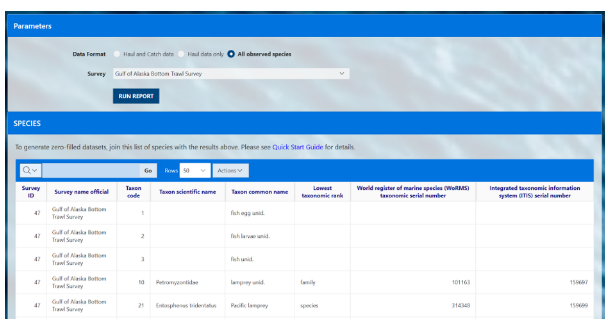


Figure 11.4.: All species observed by survey on the AFSC Groundfish and Crab Assessment Program Bottom Trawl Survey data interface on the Fisheries One Stop Shop platform.

into the "selection box" (list on the right). This can also be achieved by double clicking the option item of interest. If you accidentally select an option that you do not want to query, simply select the unwanted option from the selection box and click the left-pointing arrow (<).

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

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

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

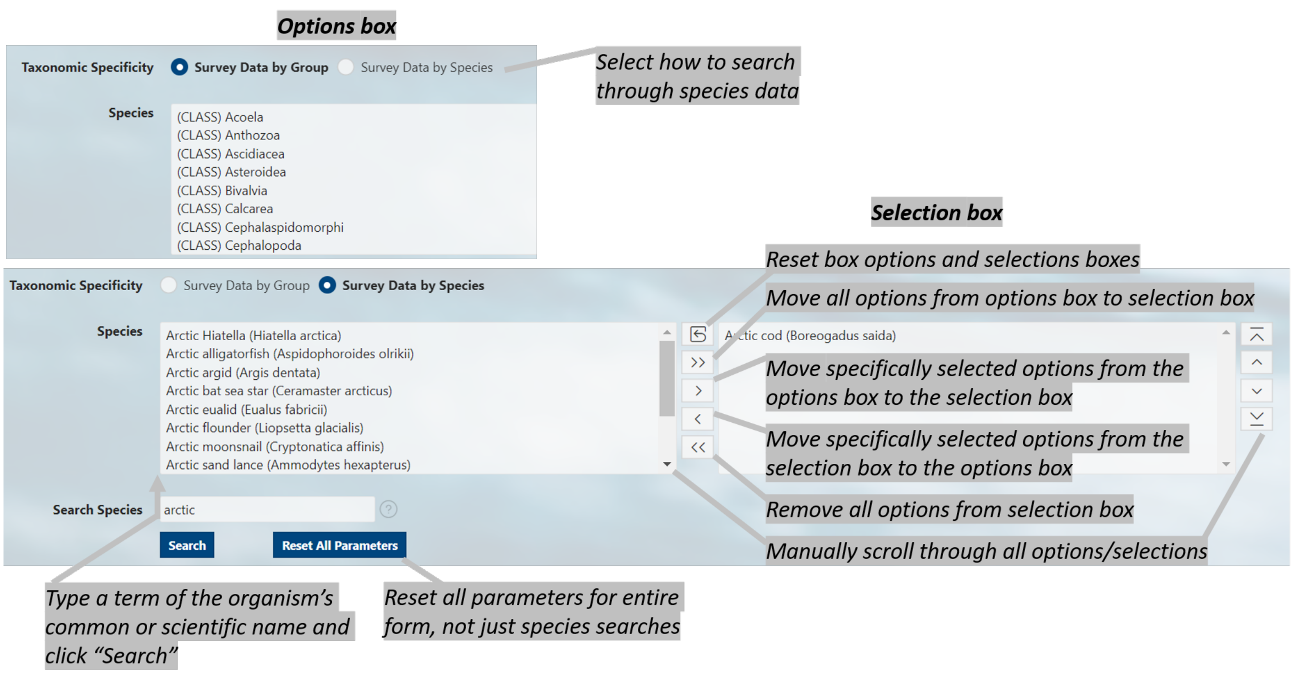


Figure 11.5.: Diagram of selection and search tools available on the FOSS platfrom.

11.3. Run report

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

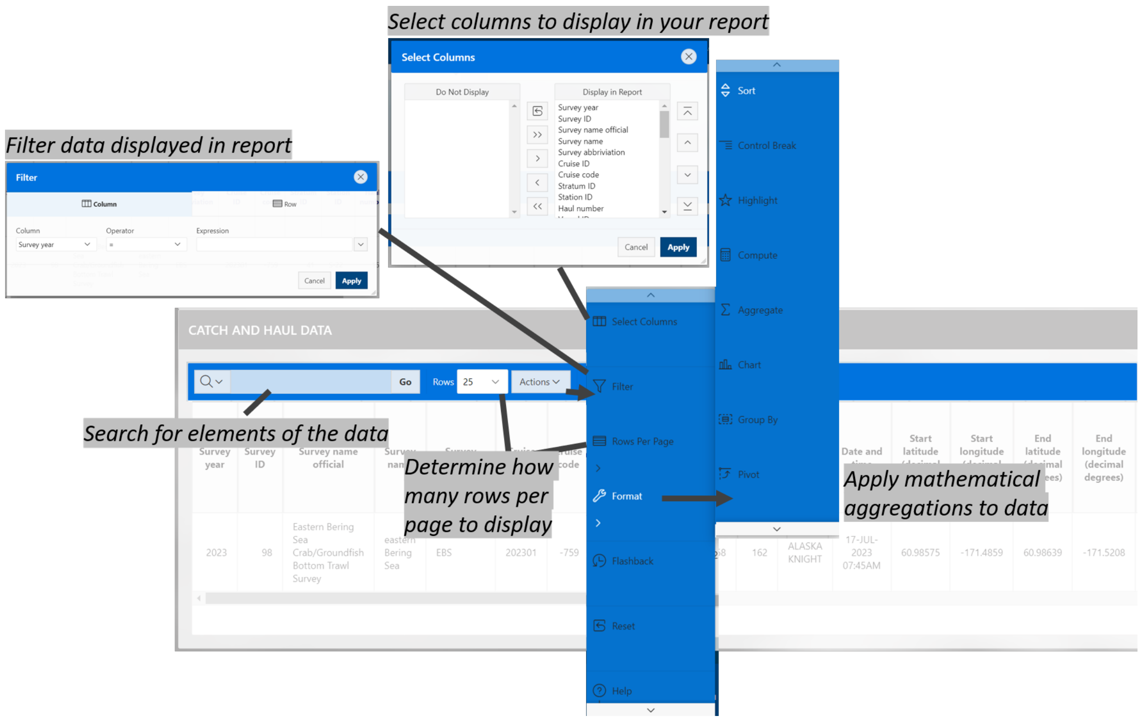


Figure 11.6.: Example data returned from running the report.

11.4. API

APIs, or Application Programming Interfaces, allows users to pull data through a IDE, or integrated development environment, like RStudio or VS Code. Explore the API pages for each of the data pages (`Haul and catch data`, `Haul data only`, `All observed species`).

12. Use data

Learn how to pull and use this data through the

- API and R programming language
- API and python programming language using the `afscgap` python package
- Oracle and R programming language (AFSC scientists only)

13. 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.

13.1. Ex. 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_survey
```

year	srvy	survey	survey_id	cruise	haul	stratum	station	vessel_-name
2002	AI	Aleutian Islands Bottom Trawl Survey	52	200201	6	722	307-63	Vesteraale
2002	AI	Aleutian Islands Bottom Trawl Survey	52	200201	6	722	307-63	Vesteraale
2002	AI	Aleutian Islands Bottom Trawl Survey	52	200201	6	722	307-63	Vesteraale

```
# 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_survey
```

year	srvy	survey	survey_id	cruise	haul	stratum	station	vessel_-name
2002	AI	Aleutian Islands Bottom Trawl Survey	52	200201	6	722	307-63	Vesteraale
2002	AI	Aleutian Islands Bottom Trawl Survey	52	200201	6	722	307-63	Vesteraale
2002	AI	Aleutian Islands Bottom Trawl Survey	52	200201	6	722	307-63	Vesteraale

13.2. Ex. Load the first 10,000 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"
```

13.3. Ex. Filter by Year

Show all the data greater than the year 2020.

```
res <- httr::GET(url = paste0(api_link, '?q={"year":{"$gt":2020}}'))
data <- jsonlite::fromJSON(base::rawToChar(res$content))

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::fit_to_width(max_width = 6) %>%
  flextable::theme_zebra() %>%
  flextable::colformat_num(x = ., j = c("year", "species_code"), big.mark = "")
```

Table 13.3.: Filter by Year.

year	srvy	stratum	species_-code	cpue_-kgkm2
2022	AI	793	80540	0.361
2022	AI	793	401	0.903
2022	AI	793	20006	1.661

13.4. Ex. 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::fit_to_width(max_width = 6) %>%
  flextable::theme_zebra() %>%
  flextable::colformat_num(x = ., j = c("year", "species_code"), big.mark = "")
```

Table 13.4.: Filter by species name.

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

13.5. Ex. Combination of year and name filters

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

```
res <- httr::GET(
  url = paste0(api_link,
               '?q={"year":{"$gt":2020},"common_name":{"$like":"%25pollock%25"}}'))
data <- jsonlite::fromJSON(base::rawToChar(res$content))

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::fit_to_width(max_width = 6) %>%
  flextable::theme_zebra() %>%
  flextable::colformat_num(x = ., j = c("year", "species_code"), big.mark = "")
```

Table 13.5.: Combination of year and name filters.

year srvy	stratum	species_-code	cpue_-kgkm2
2022 AI	793	21740	7,853.632
2022 AI	721	21740	7,235.010
2022 AI	722	21740	22,754.334

13.6. Ex. 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::fit_to_width(max_width = 6) %>%
  flextable::theme_zebra() %>%
  flextable::colformat_num(x = ., j = c("year", "species_code"), big.mark = "")
```

Table 13.6.: Combination of year, srvy, stratum.

year srvy	stratum	species_-code	cpue_-kgkm2
1989 EBS	10	66548	1.164
1989 EBS	10	69322	1.164
1989 EBS	10	43000	2.353

13.7. Ex. Visualize CPUE data in distribution map

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

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

# zero-fill data (imperfectly, but effective for this example)
res <- httr::GET(
  url = paste0(api_link, '?q={"year":2021,"srvy":"NBS"}offset=0&limit=10000'))
data_haul <- jsonlite::fromJSON(base::rawToChar(res$content))$items %>%
  dplyr::select(stratum, station, latitude_dd, longitude_dd) %>%
  dplyr::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::fit_to_width(max_width = 6) %>%
  flextable::theme_zebra()
```

Table 13.7.: Visualize CPUE data in distribution map.

stratum	station	latitude_-dd	longitude_dd	cpue_-kgkm2
81		61.66434	-172.2655	2,895.258
81		62.33740	-173.1702	1,235.545
70		62.03713	-171.6528	0.000

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

figure <- akgfmaps::make_idw_map(
  CPUE_KGHA = data$cpue_kgkm2, # calculates the same, regardless of units.
  LATITUDE = data$latitude_dd,
  LONGITUDE = data$longitude_dd,
  region = "bs.north", # Predefined EBS area
  set.breaks = "jenks", # Gets Jenks breaks from classint::classIntervals()
  in.crs = "+proj=longlat", # Set input coordinate reference system
  out.crs = "EPSG:3338", # Set output coordinate reference system
  grid.cell = c(20000, 20000))$plot + # 20x20km grid
  ggplot2::guides(fill=guide_legend(title = "Pacific cod\nCPUE (kg/km2)"))
```

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

```r
figure
```

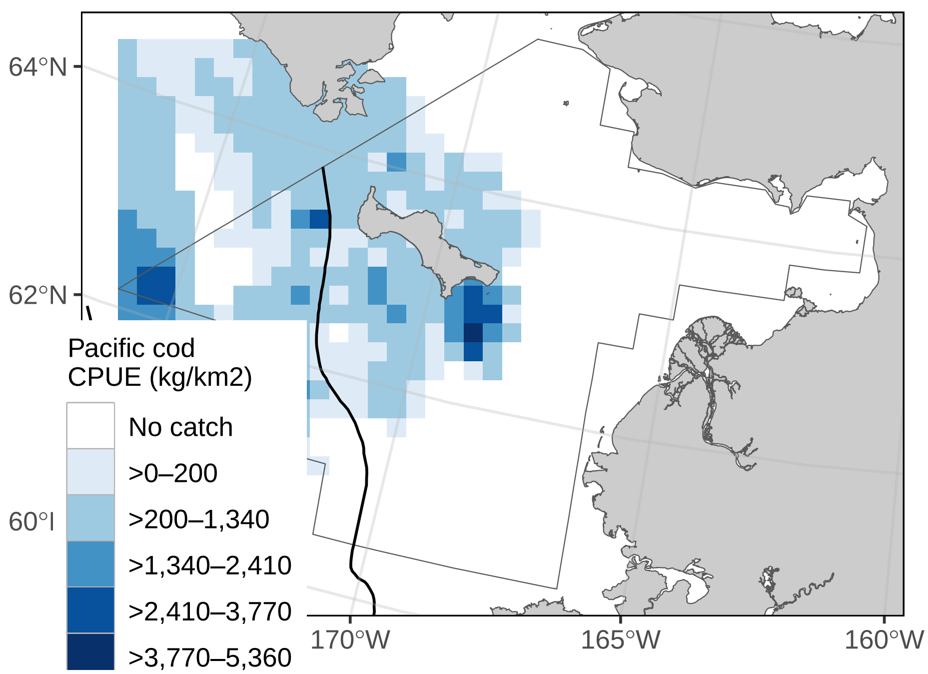


Figure 13.1.: Visualize CPUE data in distribution map.

14. Access via API and Python

14.0.1. {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.

14.0.2. Basic query

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

```
import afscgap

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

results = query.execute()
```

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

14.0.3. 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.

14.0.4. 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.

14.0.5. Load into Pandas

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

```
import pandas

import afscgap

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

```
results = query.execute()

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

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

14.0.6. 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.

14.0.7. Zero-catch inference

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

```
import afscgap

# Mapping from CPUE to count
count_by_cpue = {}

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

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

# Print the result
print(count_by_cpue)
```

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

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

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

14.0.8. More information

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

15. 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.

15.0.1. 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.

```
library(gapindex)
channel <- gapindex::get_connected()
```

15.0.2. Ex. Wholesale download data and join data in R

```
locations <- c(
  "GAP_PRODUCTS.FOSS_CATCH",
  "GAP_PRODUCTS.FOSS_HAUL",
  "GAP_PRODUCTS.FOSS_SPECIES"
)

print(Sys.Date())

error_loading <- c() # log if any tables are unable to download
for (i in 1:length(locations)){
  print(locations[i])
```

```
  a <- RODBC::sqlQuery(channel, paste0("SELECT * FROM ", locations[i], "; "))
  if (is.null(nrow(a))) { # if an error in downloading has occurred
    error_loading <- c(error_loading, locations[i])
  } else { # if no error in downloading has occurred
    write.csv(x = a,
              # change file name to be more computer file storage friendly
              here::here(paste0(tolower(gsub(
                pattern = '.',
                replacement = "_",
                x = locations[i],
                fixed = TRUE)),
                ".csv")))
  }
}
error_loading
```

Join downloaded files into presence-only table

```
# Load data
library(dplyr)
library(here)
library(readr)
catch <- readr::read_csv(file = here::here("data/gap_products_foss_catch.csv"))[,-1] # remo
haul <- readr::read_csv(file = here::here("data/gap_products_foss_haul.csv"))[,-1] # remove
species <- readr::read_csv(file = here::here("data/gap_products_foss_species.csv"))[,-1] #

dat <-
  # join haul and catch data to unique species by survey table
  dplyr::left_join(haul, catch) %>%
  # join species data to unique species by survey table
  dplyr::left_join(species) %>%
  # modify zero-filled rows
  dplyr::mutate(
    CPUE_KGKM2 = ifelse(is.null(CPUE_KGKM2), 0, CPUE_KGKM2), # just in case
    CPUE_KGHA = CPUE_KGKM2/100, # Hectares
    CPUE_NOKM2 = ifelse(is.null(CPUE_NOKM2), 0, CPUE_NOKM2), # just in case
    CPUE_NOHA = CPUE_NOKM2/100, # Hectares
    COUNT = ifelse(is.null(COUNT), 0, COUNT),
    WEIGHT_KG = ifelse(is.null(WEIGHT_KG), 0, WEIGHT_KG) )
```

Join downloaded files into zero-filled table

```
# Load data
library(dplyr)
library(here)
library(readr)
catch <- readr::read_csv(file = here::here("data/gap_products_foss_catch.csv"))[,-1] # remo
haul <- readr::read_csv(file = here::here("data/gap_products_foss_haul.csv"))[,-1] # remove
species <- readr::read_csv(file = here::here("data/gap_products_foss_species.csv"))[,-1] #

# come up with full combination of what species should be listed for what hauls/surveys
# for zero-filled data, all species caught in a survey need to have zero or non-zero row en
comb <- dplyr::full_join(
  x = dplyr::left_join(catch, haul, by = "HAULJOIN") %>%
    dplyr::select(SURVEY_DEFINITION_ID, SPECIES_CODE) %>%
    dplyr::distinct(),
  y = haul %>%
    dplyr::select(SURVEY_DEFINITION_ID, HAULJOIN) %>%
    dplyr::distinct(),
  by = "SURVEY_DEFINITION_ID",
  relationship = "many-to-many"
)

# Join data to make a full zero-filled CPUE dataset
dat <- comb %>%
  # add species data to unique species by survey table
  dplyr::left_join(species, "SPECIES_CODE") %>%
  # add catch data
  dplyr::full_join(catch, c("SPECIES_CODE", "HAULJOIN")) %>%
  # add haul data
  dplyr::full_join(haul) %>% # , c("SURVEY_DEFINITION_ID", "HAULJOIN")
  # modify zero-filled rows
  dplyr::mutate(
    CPUE_KGKM2 = ifelse(is.null(CPUE_KGKM2), 0, CPUE_KGKM2),
    CPUE_KGHA = CPUE_KGKM2/100, # Hectares
    CPUE_NOKM2 = ifelse(is.null(CPUE_NOKM2), 0, CPUE_NOKM2),
    CPUE_NOHA = CPUE_NOKM2/100, # Hectares
    COUNT = ifelse(is.null(COUNT), 0, COUNT),
    WEIGHT_KG = ifelse(is.null(WEIGHT_KG), 0, WEIGHT_KG) )
```

15.0.3. Ex. Join data using Oracle

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

```
SELECT
hh.YEAR,
hh.SRVY,
hh.SURVEY,
hh.SURVEY_DEFINITION_ID,
hh.SURVEY_NAME,
hh.CRUISE,
hh.CRUISEJOIN,
hh.HAUL,
hh.HAULJOIN,
hh.STRATUM,
hh.STATION,
hh.VESSEL_ID,
hh.VESSEL_NAME,
hh.DATE_TIME,
hh.LATITUDE_DD_START,
hh.LONGITUDE_DD_START,
hh.LATITUDE_DD_END,
hh.LONGITUDE_DD_END,
hh.BOTTOM_TEMPERATURE_C,
hh.SURFACE_TEMPERATURE_C,
hh.DEPTH_M,
cc.SPECIES_CODE,
ss.ITIS,
ss.WORMS,
ss.COMMON_NAME,
ss.SCIENTIFIC_NAME,
ss.ID_RANK,
CASE WHEN cc.CPUE_KGKM2 IS NULL THEN 0 ELSE cc.CPUE_KGKM2 END AS CPUE_KGKM2,
CASE WHEN cc.CPUE_NOKM2 IS NULL THEN 0 ELSE cc.CPUE_NOKM2 END AS CPUE_NOKM2,
CASE WHEN cc.COUNT IS NULL THEN 0 ELSE cc.COUNT END AS COUNT,
CASE WHEN cc.WEIGHT_KG IS NULL THEN 0 ELSE cc.WEIGHT_KG END AS WEIGHT_KG,
CASE WHEN cc.TAXON_CONFIDENCE IS NULL THEN NULL ELSE cc.TAXON_CONFIDENCE END AS TAXON_CONFI
hh.AREA_SWEPT_KM2,
hh.DISTANCE_FISHED_KM,
hh.DURATION_HR,
```

```
hh.NET_WIDTH_M,
hh.NET_HEIGHT_M,
hh.PERFORMANCE
FROM GAP_PRODUCTS.FOSS_SURVEY_SPECIES sv
FULL OUTER JOIN GAP_PRODUCTS.FOSS_SPECIES ss
ON sv.SPECIES_CODE = ss.SPECIES_CODE
FULL OUTER JOIN GAP_PRODUCTS.FOSS_HAUL hh
ON sv.SURVEY_DEFINITION_ID = hh.SURVEY_DEFINITION_ID
FULL OUTER JOIN GAP_PRODUCTS.FOSS_CATCH cc
ON sv.SPECIES_CODE = cc.SPECIES_CODE
AND hh.HAULJOIN = cc.HAULJOIN
```

15.0.4. Ex. Subset data

Here, we are pulling EBS Pacific cod from 2010 - 2021:

```
# Pull data
data <- RODBC::sqlQuery(
channel = channel,
query =
"SELECT * FROM GAP_PRODUCTS.FOSS_CATCH cc
JOIN GAP_PRODUCTS.FOSS_HAUL hh
ON cc.HAULJOIN = hh.HAULJOIN
WHERE SRVY = 'EBS'
AND SPECIES_CODE = 21720 -- 'Pacific cod'
AND YEAR >= 2010
AND YEAR < 2021")

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

HAULJOIN	SPECIES_-CODE	CPUE_-KGKM2	CPUE_-NOKM2	COUNT	WEIGHT_-KG	TAXON_-CONFI-DENCE	YEAR	SRVY
-19,288	21,720	449.8301	1,876.1759	83	19.90	1	2,019	EBS
-19,252	21,720	413.4828	248.0897	12	20.00	1	2,019	EBS
-18,731	21,720	946.3481	2,592.1327	118	43.08	1	2,019	EBS

15.0.5. Ex. Find all species found in the eastern Bering Sea (EBS) survey in 2023

```
# Pull data
data <- RODBC::sqlQuery(
channel = channel,
query =
"SELECT DISTINCT
ss.COMMON_NAME,
ss.SCIENTIFIC_NAME,
ss.ID_RANK,
ss.WORMS
FROM GAP_PRODUCTS.FOSS_CATCH cc -- get species codes
LEFT JOIN GAP_PRODUCTS.FOSS_SPECIES ss -- get species info
ON cc.SPECIES_CODE = ss.SPECIES_CODE
```

```
LEFT JOIN GAP_PRODUCTS.FOSS_HAUL hh -- filter by year and survey
ON cc.HAULJOIN = hh.HAULJOIN
WHERE hh.YEAR = 2023
AND hh.SURVEY_DEFINITION_ID = 98 -- EBS survey
ORDER BY COMMON_NAME")

flextable::flextable(data[1:3,]) %>%
  # flextable::fit_to_width(max_width = 6) %>%
  flextable::theme_zebra()
```

COMMON_NAME	SCIENTIFIC_NAME	ID_RANK	WORMS
Alaska great-tellin	Megangulu luteus	species	423,511
Alaska plaice	Pleuronectes quadritu-berculatus	species	254,564
Alaska razor	Siliqua alta	species	413,689

Part VI.

Data Products & Tools

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

Table 15.1.: Survey of products developed by GAP

Product	Point of Contact GOA/AI	Point of Contact BS	Description
Data			
Finalized bottom trawl data	Ned Laman	Duane Stevenson	NOAA-NMFS-AFSC-RACE-GAP bo data that has completed the po internal QAQC process.
Data requests	Nancy Roberson	Nancy Roberson	To request a subset of the NOAA-NMFS-AFSC-RACE-GAP bo raw data or a data product.
Species codebook	Nancy Roberson	Chris Anderson	List of codes used for fish and i identified in NOAA-NMFS-AFSC- Division surveys.
Survey protocols	Nancy Roberson	Nancy Roberson	Documentation of NOAA-NMFS-AFSC-RACE-GAP g bottom trawl survey protocols.
Analysis			
Design-based indices for target species	Ned Laman	Rebecca Haehn	Standard design-based indices abundance from NOAA-NMFS-A bottom trawl survey data.
Design-based age or length composition	Ned Laman	Rebecca Haehn	Standard design-based indices composition from NOAA-NMFS-AFSC-RACE-GAP bo survey data.
Model-based indices, age comps (stock assessment), area occupied, and COG (ESP)	Cecilia O'Leary	Lewis Barnett	Spatiotemporal model-based bi abundance indices, and age co NOAA-NMFS-AFSC-RACE-GAP bo survey data.
Annual bottom and surface temperature summary (ESR, stock assessment)	Cecilia O'Leary	Sean Rohan & Lewis Barnett	Summary metrics for bottom tr and surface temperatures relat baseline.

Product	Point of Contact GOA/AI	Point of Contact BS	Description
Bering Sea cold pool index and temperature data products (ESR, ESP, stock assessment)	-	Sean Rohan & Lewis Barnett	Create annual temperature rast EBS, calculate the EBS cold pool temperature data products, and visualizations.
Annual fish condition (ESR)	Cecilia O'Leary	Bianca Prohaska & Sean Rohan	Groundfish morphometric cond the Bering Sea, Aleutian Islands, Alaska.
Rockfish indices vs environmental gradients (ESR)	Alexandra Dowlin	-	GOA/AI survey trends in distribu abundance of 6 rockfishes acros environmental gradients in the
Structure-Forming Invertebrates-Habitat Areas of Particular Concern (SFI-HAPC) (ESR)	Ned Laman	-	Relative abundance of sponges, soft corals, Gorgonians, anemon Pennatulaceans in GOA and AI s
Forage fishes (ESR)	Ned Laman	-	Relative abundance of capelin, e sandfish, sand lance, and pricke and AI surveys.
Miscellaneous species (ESR)	Ned Laman	Thaddeus Buser	Relative abundance of echinode shrimp and eelpouts in GOA and
Jellies (ESR)	Ned Laman	Thaddeus Buser	Relative abundance of sea jellies surveys.
Essential fish habitat	Megsie Siple	Sean Rohan	Habitat maps for groundfish an on species distribution models. five years.
Visualization Tools			
Alaska groundfish maps (CPUE, etc.)	-	Sean Rohan	Visualization tool for the Alaska regions.
Communication			

Product	Point of Contact GOA/AI	Point of Contact BS	Description
Annual survey data report	Megsie Siple & Bethany Riggle	Emily Markowitz, Sophia Wassermann, Nicole Charriere, Chris Anderson	Alaska Fisheries Science Center Technical Memorandum summa survey progress and findings. Th available online and the latest p each survey are listed below (https://repository.library.noaa.
ADF&G report of research activities	Alexandra Dowlin	Nicole Charriere & Rebecca Haehn	Report on AI and GOA trawl sur activity inside and outside of Ala waters.
IPHC Report	-	Rebecca Haehn	
Plan team survey results presentation	Ned Laman	Duane Stevenson	NOAA-NMFS-AFSC-RACE-GAP pr findings to the North Pacific Gro Team; presentations, recordings attachments located here: https://www.npfmc.org/about-t council/plan-teams/bsai-and-go
Community highlights report	TBD	Emily Markowitz	Compilation of NOAA-NMFS-AFS survey findings for communities Alaska.
Bottom Trawl Survey Temperature and Progress Maps	Ned Laman	Emily Markowitz	Near real-time survey progress temperatures recorded during t Islands, Gulf of Alaska, and Beri Trawl Surveys.

16. Open source code

16.1. R Packages

16.1.1. akgfmaps R package

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

16.1.2. coldpool R package

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

16.1.3. akfishcondition R package

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

16.1.4. gapindex R package

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

Part VII.

Contact us

Thank you for using our data guide!

> This code is always in development. Find code used for various reports in the code releases.

This code is primarally maintained by:

Emily Markowitz (Emily.Markowitz AT noaa.gov; @EmilyMarkowitz-NOAA)

Zack Oyafuso (Zack.Oyafuso AT noaa.gov; @zoyafuso-NOAA)

Sarah Friedman (Sarah.Friedman AT noaa.gov; @SarahFriedman-NOAA)

Alaska Fisheries Science Center,

National Marine Fisheries Service,

National Oceanic and Atmospheric Administration,

Seattle, WA 98195

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

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

17. Production run notes

18. R Version Metadata

```
R version 4.3.2 (2023-10-31 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_4.3.2    fastmap_1.1.1     cli_3.6.2         tools_4.3.2
 [5] htmltools_0.5.8.1 rstudioapi_0.16.0 yaml_2.3.8        rmarkdown_2.26
 [9] knitr_1.46        jsonlite_1.8.8    xfun_0.43         digest_0.6.35
[13] rlang_1.1.3       evaluate_0.23
```

18.0.1. 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

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.

18.0.2. 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.

19. Acknowledgments

20. 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.

21. Land Acknowledgements

We would like to thank the many communities of the Bering Strait region and their members who have helped contribute to this document. The knowledge, experiences, and insights of the people of the Bering Strait region 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, including the local names used for the species covered by this document, identifying species of interest or concern that should be included in this document, and participation in an open dialog about how we can improve our collective knowledge of the ecosystem and the region.

NOAA Fisheries Alaska Fisheries Science Center's work is conducted in the waters and along the coastlines of Alaska, which include the traditional home lands and waters of the Inupiat, Yupiit, Siberian Yupiit, Unangax, Alutiiq/Sugpiaq, Eyak, Dena'ina Athabascan, Tlingit, Haida, and Tsimshian who have stewarded their lands and waters since time immemorial. We are indebted to these peoples for their wisdom and knowledge of their lands and waters.

This document was prepared in the greater Seattle area, which are the traditional lands of the Coast Salish people, including the Duwamish people, past and present. We are grateful for their continued sharing of vision, wisdom, values, and leadership.

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). (2024). *AFSC goundfish assessment program design-based production data*. NOAA Fisheries Alaska Fisheries Science Center, Goundfish Assessment Program; https://akfinbi.psmfc.org/analytics/; U.S. Dep. Commer. https://www.psmfc.org/program/alaska-fisheries-information-network-akfin

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

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

NOAA Fisheries Alaska Fisheries Science Center. (2024). *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. (2024). *AFSC goundfish assessment program design-based production data*. https://www.fisheries.noaa.gov/alaska/science-data/groundfish-assessment-program-bottom-trawl-surveys; U.S. Dep. Commer.

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

Von Szalay, P. G., Raring, N. W., Siple, M. C., Dowlin, A. N., Riggle, B. C., and Laman, E. A. and. (2023). *Data report: 2022 Aleutian Islands bottom trawl survey* (AFSC Processed Rep. No. 2023-07; p. 230). U.S. Dep. Commer. https://doi.org/10.25923/85cy-g225