Package: afscdata (via r-universe)

May 24, 2026

Title An R Package for Extraction of AFSC Fishery Data

Version 1.0.0

Maintainer The package maintainer <ben.williams@noaa.gov>

Description This package provides a folder structure and SQL queries for accessing the AKFIN and AFSC databases in support of fishery stock assessments. Additionally, archived data are held/provided to maintain consistency across assessments/assessment authors.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

URL https://afsc-assessments.github.io/afscdata

Imports DBI, getPass, here, keyring, magrittr, odbc, utils, vroom, stringi, tidytable, dplyr, lubridate, stats, stringr

Suggests knitr, rmarkdown, testthat (>= 3.0.0), tibble

Depends R (>= 3.5.0)

LazyData true

Config/testthat/edition 3

VignetteBuilder knitr

Config/pak/sysreqs make libicu-dev libsecret-1-dev unixodbc-dev

Repository https://jimianelli.r-universe.dev

Date/Publication 2026-04-22 22:32:07 UTC

RemoteUrl https://github.com/afsc-assessments/afscdata

RemoteRef HEAD

RemoteSha 96ed9cea60f38f65a27553a2b14483e2c6ce3180

Contents

`accepted_model` *copy the previous SSC accepted model as the new "base" model*

Description

copy of all base model files, adds a "README.md" file that identifies the model used (or appends the current README), predicated on using the 'afscdata::setup_folders()' function, though has an option for other folder structures. Note, this function copies everything over (e.g., retrospectives) and may take a few minutes to complete

Usage

```
accepted_model(base_year, base_model, year, folder = NULL)
```

Arguments

`base_year`	year of the base model
`base_model`	name of the base model (folder name)
`year`	current year
`folder`	if the base model is in a different location than the afscdata structure is looking for

Value

a 'base' model folder with all of the prior years inputs/outputs

Examples

```
## Not run:
accepted_model(base_year = 2020, base_model = "m18.2b", year = 2021)

## End(Not run)
```

`bsai_amak` *raw data query for the BSAI Atka mackerel assessment*

Description

raw data query for the BSAI Atka mackerel assessment

Usage

```
bsai_amak(year, off_yr = FALSE)
```

Arguments

`year`	assessment year
`off_yr`	if this is an off-year assessment change to TRUE

Value

a suite of raw data .csv files and a time stamp of when the query was done

Examples

```
## Not run:
bsai_amak(year = 2023, off_yr = TRUE)

## End(Not run)
```

`bsai_amak_catch_1977_1990` *BSAI Atka mackerel historical catch*

Description

A dataset containing historical catch data by year for 1977-1990, pulled from 2022 stock assessment data file.

Usage

```
bsai_amak_catch_1977_1990
```

Format

A data frame with 14 observations and 2 variables:

year 1977-1990

catch catch in tons

`bsai_atf` *raw data query for BSAI arrowtooth flounder*

Description

raw data query for BSAI arrowtooth flounder

Usage

```
bsai_atf(year, off_yr = FALSE)
```

Arguments

`year`	assessment year
`off_yr`	if this is an off-year assessment change to TRUE

Value

a suite of raw data .csv files and a time stamp of when the query was done

Examples

```
## Not run:
bsai_atf(year = 2022, off_yr = FALSE)

## End(Not run)
```

bsai_fhs *raw data query for BSAI flathead sole*

Description

raw data query for BSAI flathead sole

Usage

```
bsai_fhs(year, off_yr = FALSE)
```

Arguments

year	assessment year
off_yr	if this is an off-year assessment change to TRUE

Value

a suite of raw data .csv files and a time stamp of when the query was done

Examples

```
## Not run:
bsai_fhs(year = 2022, off_yr = FALSE)

## End(Not run)
```

bsai_fhs_catch_1964_1994 *BSAI flathead sole historical catch data 1964-1994*

Description

A dataset containing trawl gear catch by year for 1964-1994

Usage

```
bsai_fhs_catch_1964_1994
```

Format

A data frame with 30 observations and 2 variables:

year year of catch

catch weight of catch in 1,000 t

bsai_nrs *raw data query for BSAI northern rocksole*

Description

raw data query for BSAI northern rocksole

Usage

```
bsai_nrs(year, off_yr = FALSE)
```

Arguments

year	assessment year
off_yr	if this is an off-year assessment change to TRUE

Value

a suite of raw data .csv files and a time stamp of when the query was done

Examples

```
## Not run:
bsai_atf(year = 2022, off_yr = FALSE)

## End(Not run)
```

bsai_octopus *raw data query for BSAI octopus*

Description

raw data query for BSAI octopus

Usage

```
bsai_octopus(year)
```

Arguments

year	assessment year

Value

a suite of raw data .csv files and a time stamp of when the query was done

Examples

```
## Not run:
bsai_octopus(year = 2023)

## End(Not run)
```

`bsai_orox` *raw data query for BSAI other rockfish*

Description

raw data query for BSAI other rockfish

Usage

```
bsai_orox(year, off_yr = FALSE, catch_report = FALSE)
```

Arguments

`year`	assessment year
`off_yr`	if this is an off-year assessment change to TRUE
`catch_report`	is this a catch report year, default FALSE

Value

a suite of raw data .csv files and a time stamp of when the query was done

Examples

```
## Not run:
bsai_orox(year = 2022, off_yr = FALSE)

## End(Not run)
```

`catch_to_ss3` *Reformat Catch to SS3 structure currently only handles a single fleet, values in t*

Description

Reformat Catch to SS3 structure currently only handles a single fleet, values in t

Usage

```
catch_to_ss3(year, se = 0.01, season = 7, fleet = 1, yld_rat = NULL)
```

Arguments

year	assessment year
se	standard error in log space, typically quite small
season	numeric month for catches to be applied
fleet	numeric id of the fishery fleet
yld_rat	default NULL; else look for a vector with the projected-in year catch in output/ and replace

Value

saves a csv in output/ with the correct format

---

collapse_filters *utility function for sql queries*

Description

adds correct quotes for sql queries, nested within `sql_filter`

Usage

```
collapse_filters(x)
```

Arguments

x	variable to add quotes to

---

connect *utility function to connect to server*

Description

utility function to connect to server

Usage

```
connect(db = "akfin")
```

Arguments

db	the database schema (”akfin” or ”afsc”)

`disconnect` *utility function to disconnect from server*

Description

utility function to disconnect from server

Usage

```
disconnect(db)
```

Arguments

`db`	the database schema (e.g., akfin or afsc)

`fmp_key` *Sablefish fmp area key by location*

Description

A dataset containing area definitions

Usage

```
fmp_key
```

Format

A data frame with 923 observations and 2 variables:

loc long_lat

FMP_SUBAREA defined area for location

gap_check_bio — *compare GAP updated survey biomass to (now) retired design-based biomass estimates*

Description

compare GAP updated survey biomass to (now) retired design-based biomass estimates

Usage

```
gap_check_bio(year, species, area, type)
```

Arguments

year	current year
species	afsc species codes e.g., 30420
area	options = ai, goa, ebs, bss, nbs
type	= region, subarea, area, stat_area, stratum, inpfc, inpfc_depth, depth, reg_area_depth

Value

a list with orig values, gap values, and a basic report

Examples

```
## Not run:
out <- gap_check_bio(year = 2024, species = 30420, type = 'total', area = 'AI' )
out$report

## End(Not run)
```

goa_amak — *raw data query for the GOA Atka mackerel assessment*

Description

raw data query for the GOA Atka mackerel assessment

Usage

```
goa_amak(year)
```

Arguments

year	assessment year

Value

a suite of raw data .csv files and a time stamp of when the query was done

Examples

```
## Not run:
goa_amak(year = 2023)

## End(Not run)
```

`goa_atf` *raw data query for GOA arrowtooth founder*

Description

raw data query for GOA arrowtooth founder

Usage

```
goa_atf(year, off_yr = FALSE)
```

Arguments

`year`	assessment year
`off_yr`	if this is an off-year assessment change to TRUE

Value

a suite of raw data .csv files and a time stamp of when the query was done

Examples

```
## Not run:
goa_atf(year = 2022, off_yr = FALSE)

## End(Not run)
```

---

goa_atf_catch_1961_1990 *GOA arrowtooth flounder historical catch data*

Description

A dataset containing trawl gear catch by year for 1961-1990

Usage

```
goa_atf_catch_1961_1990
```

Format

A data frame with 30 observations and 2 variables:

Year year of catch

Catch weight of catch in 1,000 t

---

goa_dusk *raw data query for GOA dusky rockfish*

Description

raw data query for GOA dusky rockfish

Usage

```
goa_dusk(year, off_yr = FALSE)
```

Arguments

year	assessment year
off_yr	if this is an off-year assessment change to TRUE

Value

a suite of raw data .csv files and a time stamp of when the query was done

Examples

```
## Not run:
goa_dusk(year = 2022, off_yr = FALSE)

## End(Not run)
```

`goa_dusk_catch_1977_1990` *GOA dusky rockfish historical catch data*

Description

A dataset containing trawl gear catch by year for 1977-1990

Usage

```
goa_dusk_catch_1977_1990
```

Format

A data frame with 14 observations and 2 variables:

year year of catch

catch weight of catch in 1,000 t

`goa_fhs` *raw data query for GOA flathead sole*

Description

raw data query for GOA flathead sole

Usage

```
goa_fhs(year, off_yr = FALSE)
```

Arguments

`year`	assessment year
`off_yr`	if this is an off-year assessment change to TRUE

Value

a suite of raw data .csv files and a time stamp of when the query was done

Examples

```
## Not run:
goa_fhs(year = 2022, off_yr = FALSE)

## End(Not run)
```

`goa_fhs_catch_1978_1990` *GOA flathead sole historical catch*

Description

A dataset containing historical catch data by year for 1978-1990

Usage

```
goa_fhs_catch_1978_1990
```

Format

A data frame with 13 observations and 2 variables:

year 1978-1990

catch catch in tons

`goa_nork` *raw data query for GOA northern rockfish*

Description

raw data query for GOA northern rockfish

Usage

```
goa_nork(year, off_yr = FALSE)
```

Arguments

`year`	assessment year
`off_yr`	if this is an off-year assessment change to TRUE

Value

a suite of raw data .csv files and a time stamp of when the query was done

Examples

```
## Not run:
goa_nork(year = 2022, off_yr = FALSE)

## End(Not run)
```

`goa_nork_catch_1961_1992` *GOA northern rockfish historical catch data*

Description

A dataset containing trawl gear catch by year for 1961-1992

Usage

```
goa_nork_catch_1961_1992
```

Format

A data frame with 32 observations and 2 variables:

Year year of catch

Catch weight of catch in 1,000 t

`goa_octopus` *raw data query for GOA octopus*

Description

raw data query for GOA octopus

Usage

```
goa_octopus(year)
```

Arguments

`year`	assessment year

Value

a suite of raw data .csv files and a time stamp of when the query was done

Examples

```
## Not run:
goa_octopus(year = 2023)

## End(Not run)
```

`goa_pcod_catch_1977_1990`

GOA Pacific cod historical catch

Description

A dataset containing historical catch data by year for 1977-1990

Usage

```
goa_pcod_catch_1977_1990
```

Format

A data frame with 280 observations and 4 variables:

tons catch in tons

gear fishery gear type

year 1977-1990

season quarterly fishery season

`goa_pcod_larval_indices`

GOA Pacific cod historical larval index data

Description

A dataset containing larval index collections year for 1977-2022

Usage

```
goa_pcod_larval_indices
```

Format

A data frame with 92 observations and 5 variables:

year 1977-2022

seas season?

index index id

obs observation

se_log log standard error of obs

goa_pop *raw data query for GOA POP*

Description

raw data query for GOA POP

Usage

```
goa_pop(year, off_yr = FALSE)
```

Arguments

year	assessment year
off_yr	if this is an off-year assessment change to TRUE

Value

a suite of raw data .csv files and a time stamp of when the query was done

Examples

```
## Not run:
goa_pop(year = 2022, off_yr = FALSE)

## End(Not run)
```

goa_pop_catch_1960_1990 *GOA POP historical catch data*

Description

A dataset containing trawl gear catch by year for 1961-1990

Usage

```
goa_pop_catch_1960_1990
```

Format

A data frame with 30 observations and 2 variables:

year year of catch

catch weight of catch in 1,000 t

`goa_pop_fixed_fish_length_comp` *GOA POP historical fishery length comp data*

Description

A dataset containing length comp collection data by year for 1963-1977

Usage

```
goa_pop_fixed_fish_length_comp
```

Format

A data frame with 765 observations and 4 variables:

Length length in cm

value number of lengths collected

year 1963-1977

hauls_sampled number of hauls sampled

`goa_rebs` *raw data query for GOA rougheye/blackspotted rockfish*

Description

raw data query for GOA rougheye/blackspotted rockfish

Usage

```
goa_rebs(year, off_yr = FALSE)
```

Arguments

`year`	assessment year
`off_yr`	if this is an off-year assessment change to TRUE

Value

a suite of raw data .csv files and a time stamp of when the query was done

Examples

```
## Not run:
goa_rebs(year = 2021, off_yr = FALSE)

## End(Not run)
```

`goa_rebs_catch_1977_2004`

GOA rougheye/blackspotted rockfish historical catch data

Description

A dataset containing trawl and longline gear catch by year for 1977-2004

Usage

```
goa_rebs_catch_1977_2004
```

Format

A data frame with 28 observations and 2 variables:

year year of catch

catch weight of catch in t

Details

The catches from 1977-1992 were from Soh (1998), which reconstructs the catch history using an information weighting factor ( ) to combine catch histories from both survey and fishery information. The catches from 1993-2004 were constructed used observer catch data from the FMA Observer Program (Clausen et al. 2004, Appendix A). Observed catches were available from the FMA database by area, gear, and species for hauls sampled by observers. This information was used to calculate proportions of RE/BS catch by gear type. These proportions were then applied to the combined shortraker/rougheye catch from the NMFS Alaska Regional Office to yield estimates of total catch for RE/BS rockfish.

Clausen, D. M., D.H. Hanselman, J.T. Fujioka, and J. Heifetz. 2004. Gulf of Alaska shortraker/rougheye and other slope rockfish. In Stock assessment and fishery evaluation report for the groundfish resources of the Gulf of Alaska, p. 413 – 463. North Pacific Fishery Management Council, 605 W 4th Ave, Suite 306, Anchorage AK 99501. https://apps-afsc.fisheries.noaa.gov/refm/docs/2004/GOAsloperf.pdf

Soh, Sung Kwon. 1998. The use of harvest refugia in the management of shortraker and rougheye rockfish (Sebastes borealis/Sebastes aleutianus) in the Gulf of Alaska. Ph.D. Thesis – University of Washington. 194 pp.

goa_shal_flats *raw data query for GOA shallow-water flatfish*

Description

raw data query for GOA shallow-water flatfish

Usage

```
goa_shal_flats(year, off_yr = FALSE, catch_report = FALSE)
```

Arguments

year	assessment year
off_yr	if this is an off-year assessment change to TRUE
catch_report	is this a catch report year, default FALSE

Value

a suite of raw data .csv files and a time stamp of when the query was done

Examples

```
## Not run:
goa_shal_flats(year = 2022, off_yr = FALSE)

## End(Not run)
```

goa_thornyhead *raw data query for GOA thorynhead rockfish*

Description

raw data query for GOA thorynhead rockfish

Usage

```
goa_thornyhead(year, off_yr = FALSE, catch_report = FALSE)
```

Arguments

year	assessment year
off_yr	if this is an off-year assessment change to TRUE
catch_report	is this a catch report year, default FALSE

Value

a suite of raw data .csv files and a time stamp of when the query was done

Examples

```
## Not run:
goa_thornyhead(year = 2022, off_yr = FALSE)

## End(Not run)
```

---

q_bts_agecomp *query bottom trawl survey agecomps*

Description

currently only available for goa and ai total age comps (not strata)

Usage

```
q_bts_agecomp(year, species, area, db, print_sql = FALSE, save = TRUE)
```

Arguments

year	max year to retrieve data from
species	5 digit afsc species code(s) e.g., 79210 or c(79210, 90210)
area	options are 'ai' or 'goa' - can only call a single area
db	the database to query (afsc)
print_sql	outputs the sql query instead of calling the data (default: false)
save	save the file in designated folder, if FALSE outputs to global environment

Details

for the goa and ai there is a "type" switch that queries by stratum, or total - !!stratum should not be used at the moment !!. only one of these can be used at a time.

Value

saves bts agecomp data as data/raw/area_bts_age_data.csv and area_bts_age_specimen_data.csv or outputs to the global environment, also saves a copy of the SQL code used for the query and stores it in the data/sql folder.

Examples

```
## Not run:
db <- afscdata::connect("afsc")
q_bts_agecomp(year=2022, species=21921, area = "goa", db = db)

## End(Not run)
```

`q_bts_biomass` *query bottom trawl survey biomass*

Description

probably need to beef up the documentation on the ”by” switch

Usage

```
q_bts_biomass(
  year,
  species,
  area,
  type = "total",
  db,
  print_sql = FALSE,
  save = TRUE
)
```

Arguments

`year`	max year to retrieve data from
`species`	5 digit afsc species code(s) e.g., 79210 or c(79210, 90210)
`area`	options are bs (the bs+nw), bsslope, nbs, ai, goa, old_bs (was called ”standard”) - can only call a single area
`type`	”depth”, ”stratum”, ”area”, ”total”, ”inpfc”, ”inpfc_depth” - only available for goa/ai (default: ”total”) - can only use a single switch
`db`	the database to query (akfin)
`print_sql`	outputs the sql query instead of calling the data (default: false)
`save`	save the file in designated folder, if FALSE outputs to global environment

Details

six areas are available to query from bs = bering sea + northwest 1987-present (includes nw stations) - recommended bsslope = bering sea slope nbs = northern bering sea ai = aleutian islands goa = gulf of alaska old_bs = bering sea standard 1982-present (minus ~20 stations in nw) - not recommended

for the goa and ai there is a ”type” switch that queries by depth, stratum, area, total, or uses the inpfc table or inpfc by depth table only one of these can be used at a time.

Value

saves bts biomass data as data/raw/area_(type)_bts_biomass_data.csv or outputs to the global environment, also saves a copy of the SQL code used for the query and stores it in the data/sql folder.

Examples

```
## Not run:
db <- afscdata::connect("akfin")
q_bts_biomass(year=2022, species=21921, area = "goa", type = "depth", db = db)

## End(Not run)
```

`q_bts_gap_agecomp` *query GAP bottom trawl survey agecomps*

Description

query GAP bottom trawl survey agecomps

Usage

```
q_bts_gap_agecomp(year, species, area, db, print_sql = FALSE, save = TRUE)
```

Arguments

`year`	max year to retrieve data from
`species`	5 digit species code (e.g., 10110) - can place multiple in a vector c(10110, 10130)
`area`	options: ebs, ai, goa, nbs, ebs_slope, ebs_nbs, default: goa
`db`	data server to connect to (akfin)
`print_sql`	outputs the sql query instead of calling the data (default: false)
`save`	saves a file to the data/raw folder, otherwise sends output to global enviro (default: true)

Value

saves bts length data as data/raw/bts_gap_agecomp_data.csv or outputs to the global environment, also saves a copy of the SQL code used for the query and stores it in the data/sql folder.

Examples

```
## Not run:
db <- afscdata::connect()
q_bts_gap_agecomp(year=2022, species=21921, area = "goa", db = db)

## End(Not run)
```

`q_bts_gap_length` *query bottom trawl survey length data from GAP on the AKFIN server*

Description

query bottom trawl survey length data from GAP on the AKFIN server

Usage

```
q_bts_gap_length(year, species, area, db, print_sql = FALSE, save = TRUE)
```

Arguments

`year`	max year to retrieve data from
`species`	5 digit species code (e.g., 10110) - can place multiple in a vector c(10110, 10130)
`area`	ebs, nbs, ebs_slope, ebs_nbs, ai, goa - can do multiples c("bs","ai")
`db`	data server to connect to (akfin)
`print_sql`	outputs the sql query instead of calling the data (default: false) - save must be false
`save`	saves a file to the data/raw folder, otherwise sends output to global enviro (default: true)

Value

saves bts length data as data/raw/bts_length_data.csv or outputs to the global environment, also saves a copy of the SQL code used for the query and stores it in the data/sql folder.

Examples

```
## Not run:
db <- afscdata::connect()
q_bts_gap_length(year=2022, species=10110, area = "goa", db = db)

## End(Not run)
```

q_bts_gap_sizecomp *query GAP bottom trawl survey sizecomps*

Description

query GAP bottom trawl survey sizecomps

Usage

```
q_bts_gap_sizecomp(year, species, area, db, print_sql = FALSE, save = TRUE)
```

Arguments

year	max year to retrieve data from
species	5 digit afsc species code(s) e.g., 79210 or c(79210, 90210)
area	options: ebs, ai, goa, nbs, ebs_slope, ebs_nbs, default: goa
db	the database to query (akfin)
print_sql	outputs the sql query instead of calling the data (default: false)
save	save the file in designated folder, if FALSE outputs to global environment

Value

saves bts sizecomp data as data/raw/area_bts_gap_sizecomp_data.csv or outputs to the global environment, also saves a copy of the SQL code used for the query and stores it in the data/sql folder.

Examples

```
## Not run:
db <- afscdata::connect()
q_bts_gap_sizecomp(year=2022, species=21921, area = "goa", db = db)

## End(Not run)
```

q_bts_gap_specimen *query GAP bottom trawl survey specimen data from the AKFIN server*

Description

query GAP bottom trawl survey specimen data from the AKFIN server

Usage

```
q_bts_gap_specimen(year, species, area, db, print_sql = FALSE, save = TRUE)
```

Arguments

year	max year to retrieve data from
species	5 digit species code (e.g., 10110) - can place multiple in a vector c(10110, 10130)
area	options: ebs, ai, goa, nbs, ebs_slope, ebs_nbs, default: goa
db	data server to connect to (akfin)
print_sql	outputs the sql query instead of calling the data (default: false)
save	saves a file to the data/raw folder, otherwise sends output to global enviro (default: true)

Value

saves bts length data as data/raw/bts_length_data.csv or outputs to the global environment, also saves a copy of the SQL code used for the query and stores it in the data/sql folder.

Examples

```
## Not run:
db <- afscdata::connect()
q_bts_gap_specimen(year=2022, species=21921, area = "goa", db = db)

## End(Not run)
```

---

q_bts_length *query bottom trawl survey length data from the AFSC server*

Description

query bottom trawl survey length data from the AFSC server

Usage

```
q_bts_length(year, species, area, db, print_sql = FALSE, save = TRUE)
```

Arguments

year	max year to retrieve data from
species	5 digit species code (e.g., 10110) - can place multiple in a vector c(10110, 10130)
area	bs, ai, goa, or hwc, wc, hg, hbs - can do multiples c("bs","ai")
db	data server to connect to (afsc)
print_sql	outputs the sql query instead of calling the data (default: false) - save must be false
save	saves a file to the data/raw folder, otherwise sends output to global enviro (default: true)

Value

saves bts length data as data/raw/bts_length_data.csv or outputs to the global environment, also saves a copy of the SQL code used for the query and stores it in the data/sql folder.

Examples

```
## Not run:
db <- afscdata::connect("afsc")
q_bts_length(year=2022, species=10110, area = "goa", db = db)

## End(Not run)
```

---

q_bts_sizecomp *query bottom trawl survey sizecomps*

Description

currently only available for goa and ai total size comps

Usage

```
q_bts_sizecomp(year, species, area, db, print_sql = FALSE, save = TRUE)
```

Arguments

year	max year to retrieve data from
species	5 digit afsc species code(s) e.g., 79210 or c(79210, 90210)
area	options are 'ai' or 'goa' - can only call a single area
db	the database to query (afsc)
print_sql	outputs the sql query instead of calling the data (default: false)
save	save the file in designated folder, if FALSE outputs to global environment

Value

saves bts sizecomp data as data/raw/area_bts_sizecomp_data.csv or outputs to the global environment, also saves a copy of the SQL code used for the query and stores it in the data/sql folder.

Examples

```
## Not run:
db <- afscdata::connect("afsc")
q_bts_sizecomp(year=2022, species=21921, area = "goa", db = db)

## End(Not run)
```

q_bts_specimen *query bottom trawl survey specimen data from the AFSC server*

Description

query bottom trawl survey specimen data from the AFSC server

Usage

```
q_bts_specimen(year, species, area, db, print_sql = FALSE, save = TRUE)
```

Arguments

year	max year to retrieve data from
species	5 digit species code (e.g., 10110) - can place multiple in a vector c(10110, 10130)
area	bs, ai, goa, or hwc, wc, hg, hbs - can do multiples c("bs","ai")
db	data server to connect to (afsc)
print_sql	outputs the sql query instead of calling the data (default: false)
save	saves a file to the data/raw folder, otherwise sends output to global enviro (default: true)

Value

saves bts length data as data/raw/bts_length_data.csv or outputs to the global environment, also saves a copy of the SQL code used for the query and stores it in the data/sql folder.

Examples

```
## Not run:
db <- afscdata::connect("afsc")
q_bts_length(year=2022, species=21921, area = "goa", db = db)

## End(Not run)
```

q_catch *query fishery catch data from AKFIN server*

Description

Note that this function produces results that may be confidential. They can be hidden from git/GitHub by adding fish_catch_data.csv to your .gitignore file

Usage

```
q_catch(
  year,
  species,
  area,
  db,
  add_fields = NULL,
  print_sql = FALSE,
  save = TRUE
)
```

Arguments

year	max year to retrieve data from
species	species group code e.g., "DUSK" or numeric agency values e.g. c("131", "132") - must be either all 4 digit or 3 digit codes
area	fmp_area (GOA, BSAI) or fmp_subarea (BS, AI, WG, CG, WY, EY, SE) - also available (SEI, PWSI), can use all fmp_areas or all fmp_subareas, but don't mix the two
db	data server to connect to (akfin)
add_fields	add other columns to the database (must currently exist on server). "*" will return all table columns available
print_sql	outputs the sql query instead of calling the data - save must be false
save	saves a file to the data/raw folder, otherwise sends output to global enviro (default: TRUE)

Value

saves catch data as data/raw/fish_catch_data.csv or outputs to the global environment, save also saves a copy of the SQL code used for the query and stores it in the data/sql folder.

Examples

```
## Not run:
db <- afscdata::connect()
q_catch(year=2022, species="NORK", area="goa", db=db)

## End(Not run)
```

`q_catch_foreign` *query foreign catch data from the AKFIN server*

Description

something about the species names and area codes, since they are so different from everything else? pulls data from AKFIN pre1991.foreign_blend table

Usage

```
q_catch_foreign(year, species, area, db, print_sql = FALSE, save = TRUE)
```

Arguments

`year`	max year to retrieve data from
`species`	common name (e.g., "sablefish" or "all flounders") - can call multiple species
`area`	numeric area digit code - multiples is ok
`db`	data server to connect to (akfin)
`print_sql`	outputs the sql query instead of calling the data (default: false) - save must be false
`save`	saves a file to the data/raw folder, otherwise sends output to global enviro (default: true)

Value

saves catch data as data/raw/for_catch_data.csv or outputs to the global environment, also saves a copy of the SQL code used for the query and stores it in the data/sql folder.

Examples

```
## Not run:
db <- afscdata::connect()
q_catch_foreign(year=2022, species="sablefish", area = 54:57, db = db)

## End(Not run)
```

q_date — *utility function for date of data query*

Description

utility function for date of data query

Usage

```
q_date(year, loc = NULL)
```

Arguments

year	assessment year
loc	location to save file if different from default

Value

a query date file saved as `year/data/raw/data_called.txt`

q_fish_obs — *fishery observer data query*

Description

currently setup for pulling haul level detail for age or length compositions

Usage

```
q_fish_obs(year, species, area, db, print_sql = FALSE, save = TRUE)
```

Arguments

year	max year to retrieve data from
species	numeric agency values e.g. c("131", "132") - must be 3 digit codes (norpac species codes)
area	fmp_area (GOA, BSAI) or fmp_subarea (BS, AI, WG, CG, WY, EY, SE) - also available (SEI, PWSI), can use all fmp_areas or all fmp_subareas, but don't mix the two
db	data server to connect to (akfin)
print_sql	outputs the sql query instead of calling the data - save must be false
save	saves a file to the data/raw folder, otherwise sends output to global enviro (default: TRUE)

Value

saves observer data as data/raw/fish_obs_data.csv or outputs to the global environment, save also saves a copy of the SQL code used for the query and stores it in the data/sql folder.

Examples

```
## Not run:
db <- afscdata::connect()
q_fish_obs(year=2022, species=301, area="goa", db=db)

## End(Not run)
```

---

q_fish_ticket *query fish ticket data*

Description

query fish ticket data

Usage

```
q_fish_ticket(
  year,
  species,
  area,
  db,
  add_fields = NULL,
  print_sql = FALSE,
  save = TRUE
)
```

Arguments

year	max year to query through (and location to save results)
species	species group code e.g., "DUSK"
area	fmp_area (GOA, BSAI) or fmp_subarea (BS, AI, WG, CG, WY, EY, SE) - also available (SEI, PWSI), can use all fmp_areas or all fmp_subareas, but don't mix the two
db	data server to connect to (akfin)
add_fields	add other columns to the database (must currently exist on server). "*" will return all table columns available
print_sql	outputs the sql query instead of calling the data - save must be false
save	saves a file to the data/raw folder, otherwise sends output to global enviro (default: TRUE)

Value

saves fish ticket data as data/raw/fishticket_data.csv or outputs to the global environment, save also saves a copy of the SQL code used for the query and stores it in the data/sql folder.

Examples

```
## Not run:
q_fish_ticket(year=2022, species="NORK", area="goa", db=db)

## End(Not run)
```

---

q_fsh_length *query fishery length data from the AKFIN server*

Description

query fishery length data from the AKFIN server

Usage

```
q_fsh_length(
  year,
  species,
  area,
  db,
  add_fields = NULL,
  print_sql = FALSE,
  save = TRUE
)
```

Arguments

year	max year to retrieve data from
species	3 digit species codes (e.g., 201) - can place multiple in a vector c(201, 131)
area	bs, ai, goa, or hwc, wc, hg, hbs - can do multiples c("bs","ai")
db	data server to connect to (akfin)
add_fields	add other columns to the database (must currently exist on server). "*" will return all table columns available
print_sql	outputs the sql query instead of calling the data (default: false)
save	saves a file to the data/raw folder, otherwise sends output to global enviro (default: true)

Value

saves fishery length data as data/raw/fish_length_data.csv or outputs to the global environment, also saves a copy of the SQL code used for the query and stores it in the data/sql folder.

Examples

```
## Not run:
db <- afscdata::connect("akfin")
q_fsh_length(year=2022, species=301, area="goa", db=db)

## End(Not run)
```

`q_fsh_specimen` *query fishery specimen data from the AKFIN server*

Description

query fishery specimen data from the AKFIN server

Usage

```
q_fsh_specimen(
  year,
  species,
  area,
  db,
  add_fields = NULL,
  print_sql = FALSE,
  save = TRUE
)
```

Arguments

`year`	max year to retrieve data from
`species`	3 digit species codes (e.g., 201) - can place multiple in a vector c(201, 131)
`area`	bs, ai, goa, or hwc, wc, hg, hbs - can do multiples c("bs","ai")
`db`	data server to connect to (akfin)
`add_fields`	add other columns to the database (must currently exist on server). "*" will return all table columns available
`print_sql`	outputs the sql query instead of calling the data (default: false)
`save`	saves a file to the data/raw folder, otherwise sends output to global enviro (default: true)

Value

saves fishery specimen data as data/raw/fish_specimen_data.csv or outputs to the global environment, also saves a copy of the SQL code used for the query and stores it in the data/sql folder.

Examples

```
## Not run:
db <- afscdata::connect("akfin")
q_fsh_specimen(year=2022, species=301, area = "goa", db = db)

## End(Not run)
```

`q_gap_biomass` *query GAP products bottom trawl survey biomass note: pay attention to the 'design_year' column, particularly for region or depth queries as there may be data duplicates*

Description

six areas are available to query from bs = bering sea + northwest 1987-present (includes nw stations) - recommended bsslope = bering sea slope nbs = northern bering sea ai = aleutian islands goa = gulf of alaska old_bs = bering sea standard 1982-present (minus ~20 stations in nw) - not recommended

Usage

```
q_gap_biomass(
  year = 2024,
  species = 10110,
  area = "goa",
  type = "region",
  design_yr = 2025,
  db,
  print_sql = FALSE,
  save = TRUE
)
```

Arguments

`year`	max year to retrieve data from, and default folder location
`species`	5 digit afsc species code(s) e.g., 79210 or c(79210, 90210)
`area`	options are bs (the bs+nw), bsslope, nbs, ai, goa, old_bs (was called "standard") - can only call a single area
`type`	the goa and ai have: region, reg_area, nmfs_stat_area, stratum, inpfc, inpfc_depth, depth; the bs has: region, subarea, stratum, depth; the bsslope has: region, subarea, stratum; the nbs has region, stratum - can only use a single type (default: "region")
`design_yr`	the survey design year, using multiple will double (or more your values): default 2025
`db`	the database to query (akfin)
`print_sql`	outputs the sql query instead of calling the data (default: false)
`save`	save the file in designated folder, if FALSE outputs to global environment

Value

saves bts biomass data as data/raw/(area)_(type)_bts_biomass_data.csv or outputs to the global environment, also saves a copy of the SQL code used for the query and stores it in the data/sql folder.

Examples

```
## Not run:
db <- afscdata::connect()
q_gap_biomass(year=2024, species=10110, area="bs", type="region", db=db)

## End(Not run)
```

---

q_lls_length *query nmfs longline survey raw length frequencies at the depth stratum and station level*

Description

longline database documentation available on akfin

Usage

```
q_lls_length(
  year,
  species,
  area = c("goa", "bs", "ai"),
  use_historical = FALSE,
  db,
  print_sql = FALSE,
  save = TRUE
)
```

Arguments

year	max year to retrieve data from
species	5 digit afsc/race species code(s) e.g., 20510 for sablefish or c(30576, 30050) for both shortraker and rougheye/blackspotted
area	options are 'goa', 'bs', 'ai', or a combo. default=c('goa', 'bs', 'ai')
use_historical	T/F include historical Japanese survey data in the results (default: false)
db	the database to query (akfin)
print_sql	outputs the sql query instead of calling the data (default: false) - save must be false
save	save the file in designated folder, if FALSE outputs to global environment

Details

primarily used in stock assessments for calculating sample size tables

sex-specific lengths are collected for sablefish, giant grenadier, spiny dogfish, pacific cod, and greenland turbot (starting in 2021!). sex codes: 1=male, 2=female, 3=unknown

source table on akfin: lls_length_summary_view

Value

saves lls length data as data/raw/lls_length_data.csv or outputs to the global environment. also saves a copy of the SQL code used for the query and stores it in the data/sql folder.

Examples

```
## Not run:
# raw sablefish length frequencies in 1988 (year when the domestic survey started)
# in the gulf of alaska

db <- connect("akfin")
q_lls_length(year=1988, species=20510, area="goa", db=db)

## End(Not run)
```

---

q_lls_rpn *query nfms longline survey relative population numbers or weights*

Description

longline database documentation available on akfin

Usage

```
q_lls_rpn(
  year,
  species,
  area = c("goa", "bs", "ai"),
  by = "fmpsubarea",
  use_historical = FALSE,
  db,
  print_sql = FALSE,
  save = TRUE
)
```

Arguments

year	max year to retrieve data from
species	5 digit afsc/race species code(s) e.g., 20510 for sablefish or c(30576, 30050) for both shortraker and rougheye/blackspotted
area	options are 'goa', 'bs', 'ai', or a combo. default=c('goa', 'bs', 'ai')
by	'depth' (stratum-level), 'geoarea' (e.g. Spencer Gully, Kodiak slope), 'councilarea' (e.g., West Yakutat, East Yakutat/Southeast), 'fmpsubarea' (e.g., Eastern Gulf of Alaska), or 'akwide' (only for sablefish). default: 'fmpsubarea' - can only call a single area. note that variances are not available at the depth stratum level (by = 'depth')
use_historical	T/F include historical Japanese survey data in the results (default: false)
db	the database to query (akfin)
print_sql	outputs the sql query instead of calling the data (default: false) - save must be false
save	save the file in designated folder, if FALSE outputs to global environment

Details

variables of interest: cpue = numbers per skate (1 skate = 45 hooks); relative population numbers (rpns) = area-weighted cpue (area estimates are defined by depth strata and geographic area), relative population weights (rpw) = rpn multiplied by mean fish weight, which is calculated using an allometric relationship and the mean length of fish collected in a given strata and geographic area.

methods for variance estimation of these indices are documented on p 26 the 2016 sablefish safe pdf (p 350 of the goa safe).

the time series starts for the domestic longline survey in the bs and ai starts in 1996, and there are historical data available in the bs/ai from the cooperative Japanese/U.S. survey. in the modern/domestic survey, the eastern ai are surveyed in even years, and the bs is surveyed in odd years. the longline survey does not sample the western ai. estimates in nw and sw ai are based on fixed ratios in the ne and se ai, respectively, from historical cooperative Japanese/U.S. surveys in 1979-1994.

sablefish: includes data from strata 3-7 (depths 201-1000 m) and rpns are adjusted for sperm whale depredation

all other species: includes data from all strata (depths 151-1000 m)

source tables on akfin: lls_area_stratum_rpn lls_area_stratum_rpn_depred lls_area_rpn_all_strata lls_area_rpn_3_to_7 lls_area_rpn_3_to_7_depred lls_council_sablefish_area_all_strata lls_council_sablefish_area_3_to_7_depred lls_fmp_subarea_all_strata lls_fmp_subarea_3_to_7_depred lls_ak_wide_3_to_7_depred

Value

saves lls rpn/rpw data as data/raw/lls_rpn_(by)_data.csv or outputs to the global environment. also saves a copy of the SQL code used for the query and stores it in the data/sql folder.

Examples

```
## Not run:
db <- connect("akfin")
# sablefish domestic survey rpn/rpw time series (1990-2022) by goa fmp subarea (wgoa,
# cgoa, egoa). note that sablefish rpns are corrected for sperm whale
# depredation and only include data from strata 3-7

q_lls_rpn(year=2022, species=20510, area='goa', by='fmpsubarea', db=db, save = FALSE)

# pcod domestic survey rpn/rpw time series by bering sea geographic area and
# depth stratum. note that the domestic bs time series is odd years starting in
# 1997 (ai starts in 1996).

q_lls_rpn(year=2022, species=21720, area='bs', by='depth', db=db, save=FALSE)

## End(Not run)
```

---

q_lls_rpn_length *query nfms longline survey rpn-weighted length frequencies*

Description

longline database documentation available on akfin

Usage

```
q_lls_rpn_length(
  year,
  species,
  area = c("goa", "bs", "ai"),
  by = "fmpsubarea",
  use_historical = FALSE,
  db,
  print_sql = FALSE,
  save = TRUE
)
```

Arguments

year	max year to retrieve data from
species	5 digit afsc/race species code(s) e.g., 20510 for sablefish or c(30576, 30050) for both shortraker and rougheye/blackspotted
area	options are 'goa', 'bs', 'ai', or a combo. default=c('goa', 'bs', 'ai')
by	'depth' (stratum-level), 'geoarea' (e.g. Spencer Gully, Kodiak slope), 'councilarea' (e.g., West Yakutat, East Yakutat/Southeast), 'fmpsubarea' (e.g., Eastern Gulf of Alaska), or 'akwide' (only for sablefish). default: 'fmpsubarea' - can only call a single area. note that variances are not available at the depth stratum level (by = 'depth')

`use_historical` T/F include historical Japanese survey data in the results (default: false)

`db` the database to query (akfin)

`print_sql` outputs the sql query instead of calling the data (default: false) - save must be false

`save` save the file in designated folder, if FALSE outputs to global environment

Details

sablefish: includes data from strata 3-7 (depths 201-1000 m) and rpns are adjusted for sperm whale depredation

all other species: includes data from all strata (depths 151-1000 m)

sex-specific lengths are collected for sablefish, giant grenadier, spiny dogfish, pacific cod, and greenland turbot (starting in 2021!). sex codes: 1=male, 2=female, 3=unknown

results only include geographic/stratum areas that are used for rpns calc (i.e., exploitable == 1). also ' records with length = 999 have been removed. these records are present in the database to account for instances when there ' was catch in a stratum/station but no lengths collected. the ' 999 lengths ensure rpns sum properly to the area-level but should not be included in the length compositions.

available source tables on akfin: lls_length_area_3_to_7_depred lls_length_area_all_strata

Value

saves lls rpn-weighted length frequency data as data/raw/lls_rpn_length_data.csv or outputs to the global environment. also saves a copy of the SQL code used for the query and stores it in the data/sql folder.

Examples

```
## Not run:
db <- afscdata::connect("akfin")
# sablefish domestic longline survey rpn-weighted length frequencies (1990-2022) for
# goa and bsai. note that sablefish rpns are corrected for sperm whale
# depredation and only include data from strata 3-7

q_lls_rpn_length(year=2022, species=20510, area=c('bsai','goa'), db=db, save=FALSE)

# pcod domestic longline survey rpn-weighted length frequencies (1997-2022,
# odd years only) for the bering sea

q_lls_rpn_length(year=2022, species=21720, area='bs', db=db, save=FALSE)

## End(Not run)
```

`q_lls_sable_specimen` *query nmfs longine survey specimen data (age, length, weight, sex, maturity). only available for sablefish!*

Description

longline database documentation available on akfin

Usage

```
q_lls_sable_specimen(
  year,
  area = c("goa", "bs", "ai"),
  use_historical = FALSE,
  db = akfin,
  print_sql = FALSE,
  save = TRUE
)
```

Arguments

`year`	max year to retrieve data from
`area`	options are 'goa', 'bs', 'ai', or a combo. default=c('goa', 'bs', 'ai')
`use_historical`	T/F include historical Japanese survey data in the results (default: false)
`db`	the database to query (akfin)
`print_sql`	outputs the sql query instead of calling the data (default: false) - save must be false
`save`	save the file in designated folder, if FALSE outputs to global environment

Details

source table on akfin: afsc.lls_age_view

Value

saves lls sablefish specimen data as data/raw/lls_specimen_data.csv or outputs to the global environment. also saves a copy of the SQL code used for the query and stores it in the data/sql folder.

Examples

```
## Not run:
# sablefish specimen data in 1996 (first year ages were collected in the goa domestic survey)
# in the gulf of alaska

q_lls_specimen(year=2000, area="goa", db=db)
```

```
## End(Not run)
```

`q_nontarget` *query non-target species catch estimate*

Description

non-target catch estimates by weight (or numbers)

Usage

```
q_nontarget(year, target, area, db, save = TRUE)
```

Arguments

`year`	assessment year
`target`	targeted species: 'p' = pollock-mid, 'b' = pollock-bottom, x' = rex, 'h' = shallow flats, 'k' = rockfish, 'w' = arrowtooth, 'c' = pcod, 'i' = halibut
`area`	fmp_area (GOA, BSAI) or fmp_subarea (BS, AI, WG, CG, WY, EY, SE) - also available (SEI, PWSI)
`db`	data server to connect to (akfin)
`save`	save the file in designated folder (default = T) or the global environment

Value

a csv of non-target species catch by trip target group, saved in the data/output folder

Examples

```
## Not run:
akfin = afscdaya::connect()
q_nontarget(year=2022, target="k", area="goa", db=akfin, save=FALSE)
disconnect(akfin)

## End(Not run)
```

`q_psc` *query prohibited species catch (psc) estimate*

Description

prohibited species catch (PSC) estimates reported in tons for halibut and herring, counts for salmon, crabs and other fish. Note that you can combine trip target codes c("k", "x") and regions - though results will be lumped together

Usage

```
q_psc(year, target, area, db, save = TRUE)
```

Arguments

`year`	assessment year
`target`	targeted species: 'p' = pollock-mid, 'b' = pollock-bottom, x' = rex, 'h' = shallow flats, 'k' = rockfish, 'w' = arrowtooth, 'c' = pcod, 'i' = halibut
`area`	fmp_area (GOA, BSAI) or fmp_subarea (BS, AI, WG, CG, WY, EY, SE) - also available (SEI, PWSI)
`db`	data server to connect to (akfin)
`save`	save the file in designated folder (default = T) or the global environment

Value

a csv of prohibited species catch by trip target group, saved in the data/output folder

Examples

```
## Not run:
akfin = connect()
q_psc(year=2022, target="k", area="goa", db=akfin, save=FALSE)
disconnect(akfin)

## End(Not run)
```

q_specs *Query specs*

Description

Query specs

Usage

```
q_specs(year, species, area, db, print_sql = FALSE, save = TRUE)
```

Arguments

year	for specifying the correct folder
species	species group code e.g., "DUSK", "PCOD"
area	"GOA" or "BSAI"
db	data server to connect to (akfin)
print_sql	outputs the sql query instead of calling the data - save must be false
save	saves a file to the data/raw folder, otherwise sends output to global enviro (default TRUE)

Examples

```
## Not run:
db = connect()
q_specs(year=2024, species='PCOD', area='BSAI', db=db, save=F)

## End(Not run)
```

saa_pop_60 *GOA POP historical size at age parameters data*

Description

A dataset containing size at age parameters for POP <= 1960

Usage

```
saa_pop_60
```

Format

A data frame with 1 observations and 5 variables:

linf von B parameter

k von B parameter

t0 von B parameter

a a parameter

b b parameter

---

sabl_fixed_abundance *Sablefish historical abundance data*

Description

A dataset containing both fixed gear and trawl gear catch, cpue, rpn & rpw data by year

Usage

```
sabl_fixed_abundance
```

Format

A data frame with 121 observations and 5 variables:

year year

value weight of catch in 1,000 t, or relative abundance

variable rpw, rpn, cpue, or catch

fleet japan or domestic

gear lls = longline survey, llf = longline fishery, tf = trawl fishery

---

sabl_fixed_comps *Sablefish age and length comp data*

Description

A dataset containing both age and length composition data for multiple fisheries and surveys

Usage

```
sabl_fixed_comps
```

Format

A data frame with 2010 observations and 7 variables:

year year
fleet japan or domestic
type age or length
gear lls = longline survey, tf = trawl fishery, ts = trawl survey
age if relevant, otherwise NA
length length cm if relevant, otherwise NA
comp age or length
sex male, female or NA

---

sabl_new_areas *Sablefish defining areas for age samples*

Description

A dataset containing area definitions

Usage

```
sabl_new_areas
```

Format

A data frame with 15 observations and 2 variables:

NMFS_AREA area number
AREA_NAME assignment for area number

---

sabl_vessel_lengths *vessel lengths*

Description

Vessel lengths for calculating whale depredation rates in the sablefish fishery

Usage

```
sabl_vessel_lengths
```

Format

A data frame with 7558 observations and 2 variables:

VESSEL_CODE vessel id
LENGTH length of the vessel

sabl_waa *Sablefish fixed weights at age*

Description

A dataset containing weights at age for two time blocks, by sex

Usage

```
sabl_waa
```

Format

A data frame with 90 observations and 3 variables:

waa weight at age

sex male or female

timeblock current or old

setup_folders *Setup folder structure*

Description

Creates a common folder structure for assessment data

Usage

```
setup_folders(
  year,
  dirs = c("raw", "user_input", "output", "sara", "sql"),
  tier = NULL
)
```

Arguments

year	assessment year
dirs	directories to write
tier	assessment tier to change the folders used - not currently implemented

Value

creates a designated/named folder structure

Examples

```
## Not run:
setup(2022)

## End(Not run)
```

`sharks` *raw data query for GOA/BSAI sharks*

Description

raw data query for GOA/BSAI sharks

Usage

```
sharks(year, area)
```

Arguments

`year`	assessment year
`area`	what region? eith goa or bsai

Value

a suite of raw data .csv files and a time stamp of when the query was done

Examples

```
## Not run:
sharks(year = 2023, area = "goa")

## End(Not run)
```

`sql_filter` *utility function to filter sql files*

Description

utility function to filter sql files

Usage

```
sql_filter(sql_precode = "IN", x, sql_code, flag = "-- insert species")
```

Arguments

sql_precode	change input e.g., ("=")
x	the variable to change (e.g., year)
sql_code	the sql query code...
flag	a flag in the sql code to place the precode and x in the appropriate location

Examples

```
## Not run:
.d = sql_filter(sql_precode = "<=", 2011, sql_code = .d, flag = "-- insert year")

## End(Not run)
```

---

sql_read *utility function to read sql file*

Description

utility function to read sql file

Usage

```
sql_read(x)
```

Arguments

x	the sql code to read, pulled from the top directory

Examples

```
## Not run:
.d = sql_read("fsh_catch.sql")

## End(Not run)
```

`sql_run` *utility function to run sql query*

Description

utility function to run sql query

Usage

```
sql_run(database, query)
```

Arguments

`database`	which database to connect to 'akfin' or 'afsc'
`query`	the sql query code

Examples

```
## Not run:
.d = sql_read("fsh_catch.sql")
.d = sql_filter(sql_precode = "<=", 2011, sql_code = .d, flag = "-- insert year")
.d = sql_filter(x = area, sql_code = .d, flag = "-- insert region")
.d = sql_filter(sql_precode = "IN", x = c("PEL7", "PELS"),
                   sql_code = .d, flag = "-- insert species")

afsc = DBI::dbConnect(odbc::odbc(), "afsc", UID = "afsc_user", PWD = "afsc_pwd")

sql_run(afsc, query) %>%
         vroom::vroom_write(here::here(year, 'data', 'raw', 'fsh_catch_data.csv'))
DBI::dbDisconnect(afsc)

## End(Not run)
```

Index